НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Факультет прикладної математики

Кафедра програмного забезпечення комп'ютерних систем

«На правах рукопису»
УДК 004.942

«До захисту допущено»
Науковий керівник кафедри
_____ Іван ДИЧКА
«___»_____2021р.

Магістерська дисертація

на здобуття ступеня магістра

за освітньо-науковою програмою
«Інженерія програмного забезпечення комп'ютерних
та інформаційно-пошукових систем»

зі спеціальності 121 Інженерія програмного забезпечення
на тему: «Алгоритмічно-програмний метод агрегації даних цифрових двійників»

Виконав:
студент ІІ курсу, групи КП-91мн
Пеня Олександр Романович _____

Керівник:
Доцент кафедри ПЗКС, к.т.н., доцент,
Сулема Євгенія Станіславівна _____

Консультант з нормоконтролю:
Доцент кафедри ПЗКС, к.т.н., доцент
Онай Микола Володимирович _____

Рецензент:
В.о. зав. кафедри ММСА ІПСА, к.т.н., доцент
Тимощук Оксана Леонідівна _____

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.

Студент _____

Київ – 2021 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет прикладної математики
Кафедра програмного забезпечення комп'ютерних систем

Рівень вищої освіти – другий (магістерський)

Спеціальність – 121 «Інженерія програмного забезпечення»

Освітньо-наукова програма «Інженерія програмного забезпечення комп'ютерних та інформаційно-пошукових систем»

ЗАТВЕРДЖУЮ
Науковий керівник кафедри
_____ Іван ДИЧКА
(підпис)
«__» _____2019 р.

ЗАВДАННЯ
на магістерську дисертацію
студенту Пені Олександру Романовичу

1. Тема дисертації «Алгоритмічно-програмний метод агрегації даних цифрових двійників» науковий керівник дисертації Сулема Євгенія Станіславівна, к.т.н., доцент, затверджена наказом по університету від «26» березня 2021 р. № 899-С.
2. Термін подання студентом дисертації «18» травня 2021 р.
3. Об'єкт дослідження: процес агрегації мультимодальних даних для створення цифрових двійників.
4. Предмет дослідження: метод побудови цифрового двійника системи за набором мультимодальних даних.
5. Перелік завдань, які мають бути вирішені:
   - провести аналіз існуючих методів агрегації мультимодальних даних для створення цифрових двійників, виявити їх переваги та недоліки;
   - запропонувати новий метод, що покращує деякі характеристики існуючих підходів;
   - створити програмне забезпечення, яке реалізує запропонований метод;
   - провести дослідження запропонованого методу та порівняти його з існуючими, проаналізувати отримані результати;
   - оформити звіт з роботи у вигляді документації магістерської дисертації та публікації в науковому виданні.
6. Перелік ілюстративного матеріалу:
   - алгоритм кластеризації, що враховує статистичні співвідношення;
   - алгоритм побудови та оцінки моделі;
   - використання одного і декількох наближень в різних станах;
   - діаграма моделі програмного забезпечення;
   - схема етапів методу;
   - однозначний поділ простору.

7. Перелік публікацій:

- Тези доповіді «Двохетапний метод побудови цифрових двійників за набором мультимодальних даних».
- Стаття «Аналіз залежностей у наборах темпоральних мультимодальних даних».

8. Консультанти розділів дисертації

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв
Нормоконтроль	Онай М.В., доцент кафедри ПЗКС		

9. Дата видачі завдання «15» жовтня 2019 р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів дисертації	Примітка
1.	Формулювання мети дослідження та завдання на магістерську дисертацію, ознайомлення з тематикою роботи	15.01.2020	
2.	Підбір та вивчення літератури, наукових матеріалів; визначення структури магістерської дисертації;	25.05.2020	
3.	Проведення наукового дослідження; робота над першим розділом дисертації	01.10.2020	
4.	Проведення наукового дослідження; робота над другим розділом дисертації; підготовка тез для доповіді на ПМК-2020	25.12.2020	
5.	Проведення наукового дослідження; розроблення програмного забезпечення	10.01.2021	
6.	Проведення наукового дослідження; робота над третім розділом дисертації; робота над статтею за результатами наукового дослідження	10.02.2021	
7.	Завершення роботи над основною частиною дисертації; науково-дослідна практика;	20.03.2021	
8.	Оформлення текстової і графічної частини магістерської дисертації	12.05.2021	

Науковий керівник _____ Євгенія СУЛЕМА

Студент _____ Олександр ПЕНЯ

РЕФЕРАТ

Актуальність теми. Завдяки появі великої кількості доступних та ефективних пристроїв збору інформації та їх інтеграції в розподілені системи (IoT), кількість даних, які можна отримати про різні характеристики явищ та систем є величезною. Вона агрегується у набори гетерогенних даних, які містять велику кількість як видимої (безпосередні значення), так і прихованої (неочевидні тенденції та зв'язки між величинами, їх причини) інформації та шуму.

Ефективний аналіз таких даних є ключем для розуміння досліджуваних систем та прийняття оптимальних рішень у багатьох галузях людської діяльності: виробництво, медицина, логістика, наукові дослідження, енергетика, економіка, системи моніторингу і контролю, тощо. У зв'язку з такою потребою, а також через розвиток обчислювальної техніки, математичних та алгоритмічних засобів, технологія цифрових двійників набула особливо широкого розповсюдження в останньому десятилітті.

Цифрові двійники використовуються для дослідження характеристик продуктів виробництва та моделювання роботи і критичних ситуацій у машинобудуванні, для отримання детальної інформації про стан пацієнтів із набору розрізнених аналізів у медицині, моделюють середовище для перевірки наукових гіпотез та навіть використовуються для забезпечення стабільного розвитку міст.

Втім, не зважаючи на широку розповсюдженість та практичне використання цифрових двійників, методи аналізу мультимодальних даних та побудови моделей на їх основі є предметом активних досліджень, оскільки ця задача в загальному випадку залишається нетривіальною. Покращення та дослідження продовжуються навіть для часткових випадків, для яких були запропоновані задовільні рішення.

Аналіз існуючих методів показує, що вони мають суттєві недоліки, а відповідно, шляхи покращення для досягнення більш точних, ефективних та зрозумілих результатів.

Дана робота присвячена створенню алгоритмічно-програмного методу агрегації мультимодальних даних для побудови цифрових двійників інваріантних в часі систем.

Об’єктом дослідження є процес агрегації мультимодальних даних для створення цифрових двійників.

Предметом дослідження – метод побудови цифрового двійника системи за набором мультимодальних даних.

Мета роботи полягає в підвищенні точності моделювання систем за набором мультимодальних даних, згенерованих внаслідок спостережень за умов відсутності попередніх знань про функціонування системи та зв’язки між модальностями.

Методи дослідження. В даному дослідженні використовуються методи моделювання та апроксимації, кластеризації, верифікації моделей, інтелектуального аналізу даних, обробки сигналів та статистики.

Наукова новизна полягає в тому, що було запропоновано метод поєднання мультимодальних даних, який можна адаптувати до конкретної задачі шляхом вибору реалізації його етапів і розглядати задачу моделювання як ряд менших задач і незалежно їх розв’язувати. з урахуванням специфіки задачі та досвіду дослідника.

Практичне значення отриманих результатів полягає в тому, що запропонований метод дозволяє досягти більшої точності моделювання за рахунок використання різних моделей для різних «станів» системи замість використання єдиного наближення як в існуючих методах. Створені програмні засоби дозволяють налаштувати та розширювати алгоритми, забезпечуючи гнучкість методу.

Апробація роботи. Основні положення і результати роботи були представлені та обговорювались на науковій конференції магістрантів та

аспірантів «Прикладна математика та комп'ютинг» ПМК-2020 (Київ, 18-20 листопада 2020 р.) та опубліковані у збірнику «Вісник Хмельницького національного університету».

Структура та обсяг роботи. Магістерська дисертація складається зі вступу, чотирьох розділів, висновків та додатків.

У вступі коротко описана проблематика роботи, обґрунтовується її актуальність, наводиться оцінка сучасного стану досліджуваної області.

У першому розділі детально розглядається область дослідження, наводяться існуючі методи розв'язання задачі, виявляються їх переваги, недоліки та можливості покращення.

Другий розділ присвячено створенню абстрактного методу поєднання мультимодальних даних для створення цифрового двійника системи. Наводяться етапи методу та можливі варіанти їх реалізації в конкретних випадках.

У третьому розділі наводиться проект програмного забезпечення, яке реалізує метод: виявлено та описаного вимоги до розроблюваного програмного забезпечення, обрано архітектуру ПЗ, спроектовано його компоненти, проведено аналіз та вибір засобів для розроблення ПЗ.

Четвертий розділ присвячено застосуванню запропонованого методу для моделювання енергоефективності житлових будівель за набором їх фізичних характеристик та порівняння отриманих результатів з існуючими методами створення цифрових двійників.

У висновках наводиться короткий підсумок роботи.

У додатках наводяться копії ілюстративних матеріалів та коду створеного програмного забезпечення.

Робота виконана на 82 аркушах, містить 3 додатка, посилання на список використаних літературних джерел з 31 найменування. У роботі наведено 18 рисунків та 9 таблиць.

Ключові слова: поєднання мультимодальних даних, цифровий двійник, аналіз даних, моделювання систем.

ABSTRACT

Relevance of research. Due to the emergence of a large number of available and efficient information collection devices and their integration into distributed systems (IoT), the amount of data that can be obtained about the various characteristics of phenomena and systems is huge.

It is aggregated into sets of heterogeneous data, which contain a large amount of both obvious (immediate values) and hidden (non-obvious trends and relationships between quantities, their causes) information and noise. Effective analysis of such data is the key to understanding the systems in question and making optimal decisions in many areas of human activity: manufacturing, medicine, logistics, research, energy, economics, monitoring and control systems, etc.

Due to such need, as well as the development of computer technology, mathematics and algorithms, digital twin technology has become particularly relevant in the last decade.

Digital twins are used to study the characteristics of products and model regular and critical workload situations in engineering, to obtain detailed information about the condition of patients from a set of disparate tests in medicine, to model the environment for testing of scientific hypotheses and even to ensure sustainable development of cities.

However, despite the widespread practical use of digital twins, methods of analyzing multimodal data and building models based on them are the subject of active research, as this task in general remains non-trivial. Improvements and research continue even for specific cases for which satisfactory solutions have been proposed.

Analysis of existing methods shows that they have significant shortcomings and, consequently, ways to improve to achieve more accurate, effective and interpretable results.

This work is covers the creation of multimodal data fusion method for the digital twin construction of time-invariant systems.

The object of research is the process of multimodal data fusion to create digital twins.

The subject of research is a method of creating a digital twin of the system based on a set of multimodal data.

The aim research is to increase the accuracy of modeling systems based on a set of multimodal data generated via observations in the absence of prior knowledge about the functioning of the system and the relationship between modalities.

Research methods. This work utilizes methods of modeling and approximation, clustering, model verification, data science, signal processing and statistics.

Scientific novelty. An abstract method of multimodal data fusion has been proposed that can be adapted to a specific problem by choosing the implementation of its stages. It allows for considering the modeling problem as a series of smaller problems and solving them independently taking into account the specifics of the task at hand and the experience of the researcher.

The practical significance of the obtained results is that the proposed method demonstrates greater accuracy of modeling by using different models for different "states" of the system instead of using a single approximation as in existing methods. Created software allows for configuring and extending of algorithms used in each stage, providing flexibility of the method.

Approbation of researh. The main statements and results of research were presented and discussed at the scientific conference of undergraduates and graduate students «Прикладна математика та комп'ютинг» ПМК-2020 (Kyiv, November 18-20, 2020) and published in the "Вісник Хмельницького національного університету" science magazine.

Structure of the work. The dissertation consists of an introduction, four chapters, conclusions and appendices.

In the introduction the field of study is briefly described, its relevance is demonstrated; modern state of research area is described.

The first chapter examines the area of research in detail, provides overview of existing methods for solving the problem, and identifies their advantages, disadvantages and opportunities for improvement.

The second chapter is devoted to the creation of an abstract method of multimodal data fusion to create a digital twin of the system. The stages of the method and possible options for their implementation in specific cases are given.

The third chapter deals with designing of software that implements the method. Requirements for such software are identified and described, software architecture is discussed and chosen, software components are designed and analyzed, software development tools are discussed and chosen.

The fourth chapter is devoted to the usage of the proposed method for modeling the energy efficiency of residential buildings given a set of their physical characteristics and comparing the results with existing methods of creating digital twins.

The conclusions briefly summarize the conducted research.

The appendices provide copies of illustrative materials and code of the created software.

The work is presented on 82 sheets, contains 3 appendices, 31 references. 18 figures and 9 tables are provided.

Keywords: multimodal data fusion, digital twin, data analysis, system modeling.

РЕФЕРАТ

Актуальность темы. Благодаря появлению большого количества доступных и эффективных устройств сбора информации и их интеграции в распределенные системы (IoT), количество данных, которые можно получить о различных характеристиках явлений и систем очень велико. Они агрегатируются в наборы гетерогенных данных, которые содержат большое количество как явной (непосредственные значения), так и скрытой (неочевидные тенденции и связи между величинами, их причины) информации и шума.

Эффективный анализ таких данных является ключом к пониманию изучаемых систем и принятия оптимальных решений во многих отраслях человеческой деятельности: производство, медицина, логистика, научные исследования, энергетика, экономика, системы мониторинга и контроля и тому подобное. В связи с такой необходимостью, а также ввиду развития вычислительной техники, математических и алгоритмических средств, технология цифровых двойников получила особенно широкое распространение в последнем десятилетии.

Цифровые двойники используются для исследования характеристик продуктов производства и моделирования работы и критических ситуаций в машиностроении, для получения информации о состоянии пациентов из набора разрозненных анализов в медицине, моделируют среду для проверки научных гипотез и даже используются для обеспечения стабильного развития городов.

Впрочем, несмотря на широкую распространенность и практическое использование цифровых двойников, методы анализа мультимодальных данных и построения моделей на их основе являются предметом активных исследований, поскольку эта задача в общем случае остается нетривиальной. Улучшения и исследования продолжаются даже для частных случаев, в которых были предложены качественные решения.

Анализ существующих методов показывает, что они имеют существенные недостатки, а соответственно, направления улучшения для достижения более точных, эффективных и понятных результатов.

Данная работа посвящена созданию алгоритмически-программного метода агрегации мультимодальных данных для построения цифровых двойников инвариантных во времени систем.

Объектом исследования является процесс агрегации мультимодальных данных для создания цифровых двойников.

Предметом исследования – метод построения цифрового двойника системы по набору мультимодальных данных.

Цель работы заключается в повышении точности моделирования систем по набору мультимодальных данных, сгенерированных в результате наблюдений при отсутствии предварительных знаний о функционировании системы и связей между модальностями.

Методы исследования. В данном исследовании применяются методы моделирования и аппроксимации, кластеризации, верификации моделей, интеллектуального анализа данных, обработки сигналов и статистики.

Научная новизна заключается в том, что был предложен метод слияния мультимодальных данных, который можно адаптировать к конкретной задаче путем выбора реализации его этапов и рассматривать задачу моделирования как ряд меньших задач и независимо их решать с учетом специфики задачи и опыта исследователя.

Практическое значение полученных результатов заключается в том, что предложенный метод позволяет достичь большей точности моделирования за счет использования различных моделей для различных «состояний» системы вместо использования единого приближения как в существующих методах. Созданные программные средства позволяют настроить и расширять алгоритмы, обеспечивая гибкость метода.

Апробация работы. Основные положения и результаты работы были представлены и обсуждались на научной конференции магистрантов и аспирантов «Прикладна математика і комп'ютинг» ПМК-2020 (Киев, 18-20 ноября 2020) и опубликованы в сборнике «Вісник Хмельницького Національного университету».

Структура и объем работы. Магистерская диссертация состоит из введения, четырех глав, заключения и приложений.

Во введении кратко описана проблематика работы, обосновывается ее актуальность, приводится оценка современного состояния исследуемой области.

В первом разделе подробно рассматривается область исследования, приводятся существующие методы решения задачи, выявляются их преимущества, недостатки и возможности улучшения.

Вторая глава посвящена созданию абстрактного метода слияния мультимодальных данных для создания цифрового двойника системы. Приводятся этапы метода и возможные варианты их реализации в конкретных случаях.

В третьем разделе приводится проект программного обеспечения, которое реализует метод: выявлены и описаны требования к разрабатываемому программному обеспечению, выбрана архитектура ПО, спроектированы его компоненты, проведен анализ и выбор средств для разработки ПО.

Четвертый раздел посвящен применению предложенного метода для моделирования энергоэффективности жилых зданий по набору их физических характеристик и сравнение полученных результатов с существующими методами создания цифровых двойников.

В выводах приводится краткий итог работы.

В приложениях приводятся копии иллюстративных материалов и кода созданного программного обеспечения.

Работа выполнена на 82 листах, содержит 3 приложения, ссылки на список использованных литературных источников из 31 наименования. В работе приведены 18 рисунков и 9 таблиц.

Ключеые слова: слияние мультимодальных данных, цифровой двойник, анализ данных, моделирование систем.

ЗМІСТ

СПИСОК ТЕРМІНІВ, СКОРОЧЕНЬ І ПОЗНАЧЕНЬ

Верифікація (моделі) – перевірка її істинності, адекватності; зіставлення розрахункових результатів по моделі з дійсними даними.

Кластеризація – задача групування множини об'єктів на підмножини (кластери) таким чином, щоб об'єкти з одного кластера були більш схожі один на одного, ніж на об'єкти з інших кластерів по деякому критерію.

МНК (Метод Найменших Квадратів) – математичний метод оптимізації, заснований на мінімізації суми квадратів відхилень деяких функцій від шуканих змінних.

Паттерн (шаблон) – якісне перевірене рішення типових задач в програмуванні.

Поєднання (агрегація) даних (data fusion) – процес комбінування значень, що надходять з різних джерел, наприклад, сенсорів різних типів, для більш повного опису середовища, процесу або об’єкта, що досліджується, таким чином, щоб при цьому суттєво зросла деяка інформаційна метрика.

Факторизація – розкладення об’єкта (числа, матриці, тензора) на співмножники.

Цифровий двійник (digital twin) – цифрове представлення об’єкта або системи; цифрова модель.

CSV (Comma-Separated Values) – текстовий формат представлення табличних даних. Рядок таблиці відповідає рядку тексту, що містить поля, розділені комою або крапкою з комою.

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) – алгоритм кластеризації, заснований на густині.

HOSVD (Higher Order Singular Value Decomposition) – узагальнення сингулярного розкладення для матриць більшої розмірності.

JSON (JavaScript Object Notation) – текстовий формат обміну даними, заснований на поданні об’єктів JavaScript. Дані містяться в іменованих полях структур.

MVC (Model-View-Controller) – схема поділу даних програми, інтерфейсу користувача і керуючої логіки на три окремих компоненти: модель, представлення і контролер таким чином, що модифікація кожного компонента може здійснюватися незалежно.

XML (Extensible Markup Language) – текстовий формат для подання ієрархічно структурованої інформації, в якому дані розміщуються поміж тегів – елементів, що визначають структуру.

ВСТУП

Кількість даних, які впливають більшість процесів, що відбуваються в сучасному суспільстві у всіх галузях людської діяльності невпинно зростає. Необхідність аналізу величезних масивів інформації сприяє розвитку математичних методів та технологій, які дозволяють робити висновки та приймати рішення, що ґрунтуються на результатах обробки вихідної інформації: data science, big data, відновлення інтересу до засобів машинного навчання та штучного інтелекту, розподілені системи з децентралізованою обробкою даних і, серед всього іншого, цифрові двійники.

Цифровий двійник – це електронне представлення системи, яке точно моделює деякі її властивості, які є цікавими в даному контексті. Складність таких моделей буває різною: від простих демонстраційних систем, які моделюють добре досліджені та відомі закони до надзвичайно складних програмно-апаратних комплексів, які обробляють інформацію з сотень джерел в реальному часі, виконують надзвичайно точний аналіз, адаптуються до стану своїх реальних відповідників та прогнозують подальший розвиток.

Використання цифрових двійників дозволяє проводити численні експерименти над електронною моделлю системи або процесу для отримання більшої кількості корисних фактів або перевірки різноманітних гіпотез/сценаріїв. Переваги таких моделей очевидні: вони дозволяють проводити дослідження систем без значних витрат ресурсів та без ризику економічних втрат або шкоди майну чи здоров’ю людей.

Тому подібні моделі вже активно застосовуються в багатьох галузях, але сама технологія цифрових двійників тільки розвивається та набуває популярності. Через це, зростає необхідність в методах та засобах точного аналізу великої кількості гетерогенних даних, що надходять із

різноманітних джерел та характеризують досліджувані процеси із різних боків та в різних форматах.

Складність задачі також зростає через проблеми, пов’язані з таким потоком інформації: низька якість даних (шум, викиди, похибки, відсутність синхронізації, різна природа та джерела даних) перешкоджає отриманню корисних висновків, тому попередня обробка та очищення даних можуть займати до 80% роботи в задачах аналізу. Інша проблема полягає в тому, що в зв’язках між даними прихована велика кількість корисної інформації, яка дозволяє значно покращити результати аналізу та допомогти приймати кращі рішення, які базуються на його результатах.

Для розв’язання даної проблеми використовуються методи поєднання мультимодальних даних, які і є предметом даної роботи.

1. АНАЛІЗ ПРОБЛЕМИ

1.1. Формулювання проблеми

Процеси, явища, об’єкти є складними і майже ніколи повністю не визначаються однією величиною. Навпаки, їх дослідження супроводжується вимірюванням цілого ряду параметрів, що характеризують різні аспекти, і через складність систем взаємозв’язки між цими параметрами не завжди зрозумілі або відомі. Ці спостереження утворюють колекції мультимодальних даних, обробка яких є нетривіальною задачею з ряду причин: зв’язки між досліджуваними величинами невідомі, так само невідомі фактори, які спричиняють поведінку систем, об’єм даних, що генерується моніторинговими системами може бути величезним, «сирі» вимірювання з сенсорів містять різного роду похибки та шум, вимірювання відбуваються з різною частотою, тощо. Через це виникає необхідність у створенні методів, які дозволяють виявляти закономірності в даних або принаймні використовувати їх прояв для більш точного моделювання [1-3].

Метою даного дослідження є створення методу для побудови моделей за набором мультимодальних даних, отриманих в результаті спостережень за системою – цифрових двійників. Хоча метод має спиратись переважно на самі дані як носій всієї доступної інформації про систему, важливо врахувати апріорні знання або припущення про її поведінку, оскільки хоча вони рідко доступні у повному обсязі, ця інформація може бути частково відома завдяки проведенню спостережень та досліджень або випливати із законів, що зумовлюють функціонування системи, утворюючи поєднання підходів, керованих даних та керованих моделями. Оскільки метою поєднання даних при створенні моделі є отримання додаткової інформації із зв’язків між модальностями, то використання таких припущень може підвищити точність прогнозування.

Слід зауважити, що в даній роботі поняття модель, цифрова модель та цифровий двійник використовуються як синоніми, спираючись на визначення цифрового двійника, як цифрового представлення об’єкта або системи. Хоча сучасні підходи до побудови та використання цифрових двійників часто утворюють цикл зворотного зв’язку між реальною системою та її цифровим представленням [4, 5], дане дослідження фокусується на історичних наборах даних, а можливість взаємодії з реальними системами розглядається як модифікація та напрямок подальшого дослідження.

Створений цифровий двійник має бути корисним для аналізу та отримання інформації про систему, яка моделюється, тобто виконувати прогнозування або відновлення втрачених даних, виявлення аномалій в наборах даних.

Наявний набір мультимодальних даних вважатимемо повним, тобто таким, що не містить невідомих або пропущених значень та синхронізованим за часом, втім випадкові похибки вимірів та шум допускаються. Реальні набори даних часто не є такими. В такому випадку необхідно попередньо виконати корекцію набору даних. Суть та методи корекції визначаються задачею, що розв’язуються та особливостями набору даних: від фільтрації даних до методів доповнення чи скорочення наборів даних.

1.2. Актуальність задачі

У зв’язку з розвитком обчислювальної техніки та розподілених систем, математичних та алгоритмічних засобів, а також через потреби багатьох галузей життєдіяльності, таких як промислове виробництво, медицина, продажі та логістика, енергетика, наукові дослідження, тощо, технологія цифрових двійників набула широкого розповсюдження в останньому десятилітті і входить в список найбільш перспективних за даними Gartner report.

Використання цифрових двійників дозволяє вивчати системи без безпосереднього проведення експериментів над ними для того, щоб отримати певні висновки для прийняття рішень. Це відкриває цілий ряд можливостей в дослідженнях: можливо симулювати різноманітні сценарії в великій кількості для вибору стратегій поведінки в реальних системах (в логістиці, економіці, соціології), виявляти та досліджувати критичні ситуації без ризику для майна та людей (наприклад в машинобудуванні та енергетиці), проводити симуляції середовищ, небезпечних, невідомих або складно доступних людині за допомогою вимірювань автономних станій (глибокий океан, верхні шари атмосфери, космос, токсично забруднені середовища) і багато інших застосувань цифрових моделей [6-10].

Втім, побудова ефективних цифрових двійників для реальних систем є непростою і актуальною задачею з ряду причин. По-перше, такі системи мають складну будову, тому закони, які визначають стани та поведінку системи та її компонентів дуже складні для безпосереднього моделювання, або повністю чи частково невідомі. По-друге, системи створюють велику кількість гетерогенних даних, які містять приховану інформацію у взаємозв’язках між собою та є проявом як відомих, так і невідомих властивостей системи. По-третє, під час моделювання з використанням багатьох методів складно максимально використати знання про систему або змістовно інтерпретувати отримані результати.

Дана робота присвячена гнучкому методу побудови цифрових двійників, який дозволяє за наборами мультимодальних спостережень використати якомога більше інформації, як відомої, так і невідомої досліднику при побудові моделі системи, щоб ефективніше розв’язувати поставлені задачі.

1.3. Аналіз існуючих методів поєднання даних

Розповсюдженим методом злиття даних є різноманітні тензорні розкладення, які використовують або розділені модальності окремо, або

зберігають їх поєднання в матрицях вищого порядку (тензорах), наприклад паралельний факторний аналіз, і обчислюють його розкладання на множники певної структури (HOSVD).

Додаткова інформація, прихована у неявних зв'язках між модальностями виявляється у спільних множниках або спеціальних обмеженнях на структуру співмножників.

В даному контексті розглядатимуться методи, керовані даними, а не на особливими моделями, оскільки останні передбачають розуміння принципів роботи системи, що моделюється в значному обсязі, в той час як початковим припущенням методу, що розробляється, є довільний обсяг таких відомостей: від часткових, до відсутніх.

Варто також зауважити, що дані, зібрані в результаті спостережень, є проявом характеристик системи і несуть в собі інформацію про неї. Хоча цей факт використовується при побудові методу, виведення цих закономірностей в явному вигляді є цілком іншою задачею і в даному контексті не розглядається.

1.3.1. *Методи матричної факторизації*

В методах матричної факторизації процес розглядається як деяка залежність $\bar{x} = f(\bar{s})$, де $\bar{x}$ – це результат спостереження, тобто елемент мультимодальних даних, який розглядається як точка багатовимірного простору, $\bar{s}$ – вектор прихованих (латентних) змінних, що визначають передумови процесу, наприклад, параметри, умови, сигнали та інші величини, що впливають на $\bar{x}$, при чому як правило не можуть бути виміряні безпосередньо та навіть попередньо не відомі, та f – саме перетворення, зумовлене системою, яке також може бути невідомим.

Мабуть, найбільш очевидною реалізацією такого підходу є обернена задача, де метою є отримання максимально точної апроксимації значення $\bar{s}$ та f при заданих $\bar{x}$. Відновлення f і $\bar{s}$ також можна розглядати як пошук найпростішого набору змінних, що пояснюють спостереження $\bar{x}$.

Один з найпростіших, та при цьому корисних способів подання такої задачі має вигляд

$$x_{ij} = \sum_{k=1}^{n} a_{ik} s_{jk},$$

тобто x_{ij} – це лінійна комбінація n сигналів $s_{j1}, \dots, s_{jk}$ з відповідними вагами $a_{i1}, \dots, a_{ik}$. В матричному вигляді цей вираз можна також записати у вигляді

$$X = \sum_{k=1}^{n} \bar{a}_k \bar{s}_k^T = AS^T.$$

Таким чином, маємо задачу факторизації (розкладення) матриці *X* на співмножники, при чому один з них можна інтерпретувати як розглянутий раніше набір $\bar{s}$, а інший – як складову *f* відповідно.

Проблема такого підходу полягає в унікальності розкладення. Наступне твердження справедливе для будь-якої матриці *C* відповідної розмірності, що має обернену:

$$X = AS^T = A(CC^{-1})S^T = (AC)(C^{-1}S^T).$$

Таким чином, як A, S^T, так і $AC, C^{-1}S^T$ є допустимими варіантами розкладення матриці *X*, що унеможливлює будь-яку інтерпретацію отриманих співмножників як реальних величин або сигналів. Отже, необхідно ввести додаткові обмеження на матриці-співмножники, щоб забезпечити їх унікальність. Наприклад, в математичних методах, таких як сингулярне розкладання, унікальність забезпечується спеціальною структурою матриць, такою як ортогональність або діагональність. Хоча вони (обмеження) є зручними з точки зору математичних методів, вони, по-перше, рідко застосовні в реальних системах, по-друге, порядок запису компонент у векторах не важливий, оскільки вони призводять до однакових результатів, отже, однозначної унікальності розкладення досягни неможливо за природою задачі, тому розкладення вважається унікальним, якщо матриця *C* є матрицею перестановки, і вектор-стовпці

(рядки) матриць будуть однаковими, просто в різному порядку. Так, обмеження, які накладаються на матриці в розкладанні і забезпечують унікальність випливають або з фізичних обмежень системи, які зменшують кількість ступенів свободи або її спеціальних властивостей.

Такий підхід також називається факторним аналізом. Методи, що базуються на ньому, розширюють його та розв'язують проблему унікальності включають аналіз незалежних компонент та аналіз головних компонент.

Задача *аналізу незалежних компонент* зазвичай формулюється так: для

$$\bar{x}(t) = A\bar{s}(t) + \bar{b}$$

обчислити наближення $A, \bar{s}, \bar{b}$ за наявними спостереженнями X , які максимізують незалежність компонент $\bar{s}$, тобто негаусовість їх розподілу або мінімізують певну метрику ентропії як показник незалежності. Тут $\bar{s}(t) = [s_1(t), \dots, s_n(t)] -$ це вектор n статистично незалежних випадкових величин, відомих як компоненти, $\bar{x}(t)$ – відповідні спостереження, A – вагова матриця рангу n (всі рядки лінійно незалежні), $\bar{b}$ – необов'язковий вектор шуму. Якщо покласти $[x(1), \dots, x(n)] = X,\ \ [s(1), \dots, s(n)] = Z^T$, маємо частковий випадок, що зводиться до факторного аналізу. Аналіз незалежних компонент використовує розподіл величин s, передбачений фізичними властивостями, такими як, наприклад різне розміщення сенсорів або принципи їх роботи. Разом із припущенням про статистичну незалежність компонентів це дозволяє отримати оцінки $\bar{s}(t)$, які є максимально статистично незалежними, щоб позбавитись від невизначеності [1].

Аналіз головних компонент – це непараметричний метод вилучення важливих ознак сигналу даних. Це також метод зменшення складного набору даних у спрощені структури, приховані у вихідних даних.

Зміст методу полягає у відшуканні ортогонального базису (бажано меншої розмірності), який пояснює найбільшу варіацію даних, тобто

головних компонент. Завдяки цьому розмірність зменшується без значної втрати точності через вилучення залежних елементів, які не несуть суттєвої інформації.

Хоча аналіз незалежних компонент передбачає перехід в інший базис (з невідомими базисними векторами та коефіцієнтами розкладення), то математично задача схожа на факторизацію, але алгоритм реалізації методу сильно відрізняється.

Спочатку дані нормуються, що критично для ефективного застосування методу, оскільки він є чутливим до варіації даних, тому всі величини перетворюються, забезпечуючи однакову варіацію: від значень даних віднімається середнє та отримане значення ділиться на стандартне відхилення.

Для нормованих величин обчислюється коваріаційна матриця. Для неї виконується сингулярне розкладення. Отримані в результаті власні вектори є головними компонентами початкового набору даних. Отримані компоненти пояснюють варіацію даних у порядку величини відповідних власних значень: перша компонента має найбільше власне значення і найбільше пояснює варіацію, друга – має друге значення і пояснює менше варіації і т.д.

Тепер серед отриманих компонент можна відкинути менш значущі. Для цього використовується правило Кайзера, модель зламаної тростини або оцінка за допомогою числа обумовленості, після чого нормований початковий набір даних приводиться до базису, утвореного обраними головними компонентами [11].

Окрім наведених класичних прикладів існує велика кількість інших методів факторизації та їх модифікацій, які мають різні принципи побудови та обчислення матриць-співмножників і які можна використовувати для поєднання даних, такі як сингулярне розкладення, багатовимірне шкалування, розкладання невід’ємних матриць, а також їх узагальнення для тензорів [1].

Методи факторного аналізу використовувались десятки років для аналізу широкого спектру даних, їх успіх багато в чому завдячує простоті їхньої основної ідеї та факту що існують дуже ефективні алгоритми, що дають задовільні результати. Однак слід зауважити, що на практиці багато спостережень можна краще пояснити іншими типами моделей які не обмежуються розкладанням в лінійні комбінації, статистичною незалежністю або навіть факторизацією матриць. Для досягнення унікальності також використовуються інші властивості, наприклад, невід'ємність, розрідженість, гладкість.

Покращити точність та досягти унікальності розкладання також за допомогою використання інших алгебраїчних структур, в тому числі тензорів – матриць вищого порядку.

Хоча методи факторизації використовуються для поєднання даних та отримання додаткової інформації про систему, задача, яку вони розв'язують по суті є оберненою: зафіксовані спостереження є результатом поєднання прихованих змінних завдяки невідомому перетворенню [12]. Ця ідея проілюстрована на рис. 1.1 з відповідними позначеннями: 1 – приховані змінні, 2 – перетворення та побічні ефекти, що виконують поєднання (невідомих) даних, 3 – зафіксовані спостереження, що є результатом поєднання, 4 – отримані наближення, виділення компонентів та поєднання даних водночас.

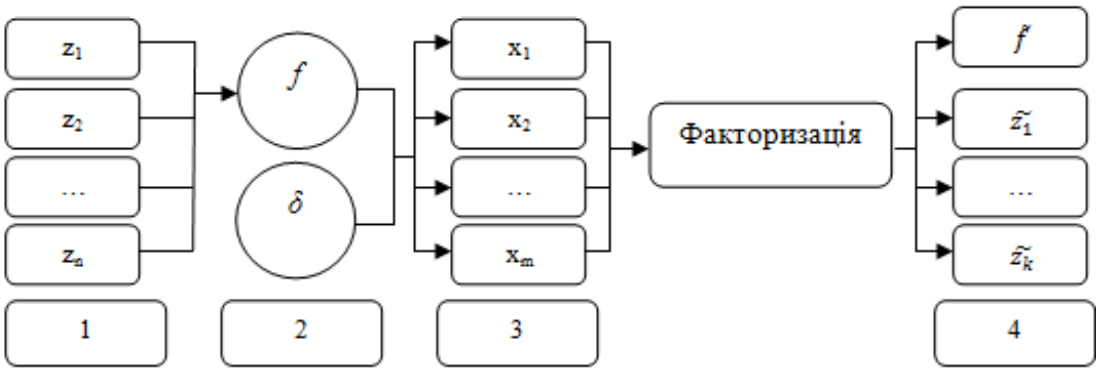


Рис. 1.1. Поєднання даних за допомогою факторизації: пряма та обернена задача

1.3.2. *Методи, що використовують поєднаний аналіз*

Наступний крок у розвитку методів розкладання пов’язаний з використанням поєднання декількох наборів даних за спільними модальностями. До таких методів відносяться паралельний факторний аналіз, поєднаний аналіз незалежних компонент та поєднані тензорні розкладення.

Основна суть таких методів полягає в розгляді декількох наборів спостережень, поєднаних деякими модальностями, як частину простору всіх можливих спостережень в базисі прихованих параметрів (рис. 1.2). Тоді задача поєднання даних та моделювання системи зводиться до відшукання точок цього простору, яких немає в наборах даних.

Їх реалізація передбачає обчислення параметрів моделі, схожої з факторним аналізом, але з тензорами вищого порядку або більшою кількістю співмножників. Таким чином можна провести аналіз для наборів даних, обмеження для яких значно слабші, ніж ті, які вимагаються для досягнення унікального матричного розкладення.

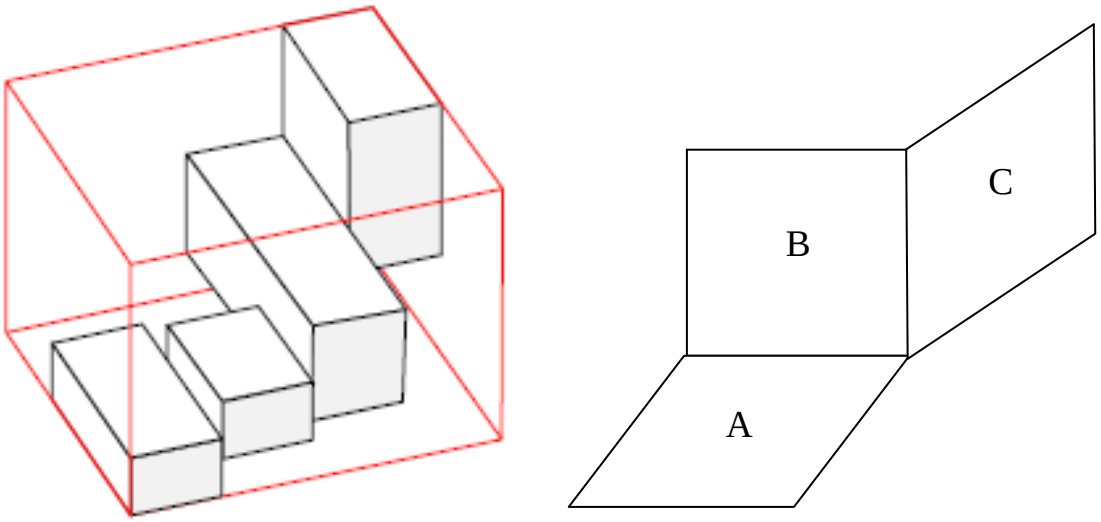


Рис. 1.2. Поєднання наборів даних за окремими модальностями в багатовимірному просторі

Аналіз незалежних векторів є узагальненням методу аналізу незалежних компонент, яке використовує поєднання окремих наборів даних, тому формулювання задачі дуже подібне: за заданими K наборами спостережень $X_i, i = \overline{1..K}$, кожен з яких є лінійною комбінацією L компонент:

$$\bar{x}^{(k)}(t) = A^{(k)}\bar{s}(t), k = \overline{1..K}$$

обчислити наближення $A^{(k)}$ (в загальному випадку K різних матриць) та $\bar{s}$.

Дану задачу можна розглядати як набір K окремих задач аналізу незалежних компонент, які потім необхідно «накласти» одна на одну, тобто встановити відповідність між компонентами в окремих результуючих наближеннях, що як правило розглядається як окрема задача класифікації. Більш ефективною є реалізація, яка використовує статистичну залежність між величинами в різних наборах даних (всередині одного набору величини мають бути незалежні, як і для аналізу незалежних компонент), оскільки такий підхід одразу призводить до результату, з встановленою відповідністю ніж компонентами [1].

Алгоритми аналізу незалежних векторів дозволяють використання лінійної або нелінійної залежності між величинами в різних наборах даних. Також, існують модифікації, застосовні до випадків, у яких існує залежність між спостереженнями всередині одного набору даних.

Відповідно, паралельний факторний аналіз є узагальненням аналізу головних компонент для багатовимірних наборів даних або поєднань декількох наборів за спільними модальностями.

Паралельний факторний аналіз базується на математичній моделі, яка представляє взаємодію вимірів, в яких слід аналізувати вхідні дані. Кожне вхідне значення може бути пов’язане з індексом кожного з вимірів, утворюючи N-вимірний масив спостережень X з елементами $x_{i_1 i_2 \dots i_N}$, для якого потім виконується розкладення.

Необхідно обрати кількість компонентів K та ініціалізувати матриці $A_1,\ldots,A_N$ (зазвичай ініціалізуються невеликими додатними значеннями) розмірності $i_1 \times K,\ i_2 \times K,\ \ldots,\ i_N \times K$ відповідно.

Тоді модель розкладення має вигляд

$$x_{i_1 i_2 \ldots i_N} = \sum_{j=1}^{K} a_{i_1 j}\, a_{i_2 j}\, \ldots a_{i_N j} + \varepsilon_{i_1 i_2 \ldots i_N},$$

де $\varepsilon_{i_1 i_2 \ldots i_N}$ позначає похибку моделювання, яку необхідно мінімізувати, наприклад, за допомогою алгоритму змінних найменших квадратів:

1. Обрати кількість компонентів K та ініціалізувати матриці.
2. Для $i = \overline{1..N}$:
   2.1. Вважаючи всі матриці, окрім A_i постійними, обчислити A_i, мінімізуючи похибку моделювання ε методом найменших квадратів.
3. Повторювати крок 2 до збіжності алгоритму (збіжність визначається малою зміною матриць або похибки між ітераціями).

Вибір кількості компонентів K є неочевидним, і методів, що визначають її точне значення не існує, хоча запропоновано декілька підходів, один з яких передбачає поступове збільшення K, доки не буде отримана модель з задовільною точністю. Якщо K вибрано замалим, не всі закономірності у вхідних даних будуть визначені. Якщо K вибрано занадто великим, шум все більше впливає на модель, що призводить до виділення залежних компонент, які не несуть корисної інформації [13].

Значною перевагою моделі паралельного факторного аналізу є унікальність розкладення. Якщо дані дійсно N-лінійні, то відповідні базові компоненти будуть знайдені, за умов правильного вибору K і зашумлення даних в наборі не є занадто великим.

Так само, як і з методами факторизації, на цих класичних методах ґрунтуються численні модифікації, узагальнення, реалізації алгоритмів, які оперують різними припущеннями про вихідні дані та більш пристосовані

для окремих класів задач. Серед них поєднане канонічне поліадичне розкладення, аналіз незалежних підпросторів, тензорне розкладення Такера та його модифікації, поєднані тензорні розкладення, і т.д.

Точні умови унікальності поєднаних тензорних розкладень різної розмірності залишаються темою активних досліджень. Втім результати використання таких методів в окремих випадках призводять до кращих результатів, ніж використання двовимірних лінійних моделей, хоча умови та границі їх ефективного застосування ще не до кінця вивчені.

1.3.3. *Імовірнісні підходи до поєднання даних*

Невизначеність в тій чи іншій формі присутня в процесах спостереження, передачі, аналізу та поєднання даних. Потрібно надати чітке вимірювання цієї невизначеності, щоб забезпечити поєднання сенсорної інформації, щоб процес злиття даних був ефективним, і можна було точно прогнозувати стан системи або об'єкта, що досліджується. Більшість таких методів подання невизначеності базуються на використанні імовірнісних моделей, оскільки ці моделі забезпечують потужний засіб опису невизначеності у багатьох випадках. Ця концепція вписується в ідеї поєднання інформації та прийняття рішень. Звичайно, певні практичні аспекти проблеми часто вимагають використання якогось іншого методу моделювання невизначеності, наприклад, концепцій на основі нечіткої логіки та їх використання у фільтрації та поєднанні даних [14, 15].

Хоча імовірнісні моделі підходять для багатьох ситуацій, ці моделі не можуть охопити всю інформацію, необхідну для визначення та опису спостереження, аналізу та поєднання даних, наприклад, евристичні знання експертів в області знань, для якої теорія нечіткої логіки добре підходять.

Практика відтворення відсутніх даних виникла з ідеї, що кожну відсутню точку даних можна замінити найкращою оцінкою того, яким було б спостережуване значення якби воно було наявним. Таким чином,

відтворення даних є привабливою стратегією, оскільки вона дає повний набір даних для подальшого аналізу. Початкова реалізація цієї ідеї використовувала одне відтворення, в якому відсутні дані замінювались єдиним значенням, яке часто оцінювалось середніми значеннями з наявних даних або за допомогою оцінок регресії. Однак ці значення відтворення часто виявляються поганими оцінками неспостережуваних значень і дають необ'єктивні оцінки параметрів.

На відміну від цього, методи множинного обчислення вважаються дуже ефективними та гнучкими інструментами для аналізу даних із відсутніми значеннями. На відміну від одинарних обчислень, де кожна відсутня точка даних заповнюється одним розрахунковим правдоподібним значенням, при множинному обчисленні кожне відсутнє значення замінюється кількома ймовірними значеннями. Загальна ідея цих численних методів обчислення полягає в тому, що для кожного відсутнього значення у поєднаному наборі даних мається кілька значень, у відповідності до деяких припущень про розподіл відсутніх даних, які можуть бути відтворені за допомогою явної баєсівської моделі.

Баєсівська модель передбачає перехід від одного імовірнісного розподілу значень (попереднього або апріорного), що визначається сукупністю об’єктивних та суб’єктивних чинників, такими як спостереження, висновки та припущення експерта, певні очікування, тощо до іншого розподілу (апостеріорного) за рахунок використання додаткової інформації згідно теореми Баєса.

Припустимо, задано простір гіпотез S і набір даних X, тоді ми можна визначити імовірності: $P(s)$ – імовірність того, що s є правильною гіпотезою перед використанням будь-яких даних (апріорна імовірність), $P(X)$ – імовірність отримання даних X та $P(X|s)$ – імовірність появи даних X, за умови, що s правильна. Теорема Баєса пов'язує дані імовірності:

$$P(s|X) = \frac{P(X|s) \cdot P(s)}{P(X)}.$$

Метод, знаходження гіпотези з максимальною $P(s|X)$, називається методом максимальної апостеріорної імовірності: $H_{map} = arg\{max_s(P(s|X))\}$.

Баєсівська модель використовує суб'єктивні ймовірності: вона обчислює імовірність істинності гіпотези за наявних спостережень, можна включити апріорне знання про імовірність істинності гіпотези, при чому знання функцій розподілу імовірності не потребується.

Суть цього результату полягає в інтерпретації функцій розподілу. Нехай задано дві випадкові змінні, x та s Припустимо, що необхідно визначити різні ймовірності значень невідомого стану x. Існують попередні уявлення про очікувані значення x, і вони кодуються у апріорній імовірності $P(x)$. Якщо необхідно отримати більше інформації про x, потрібно зробити певні вимірювання z з припущенням, що z якось пов'язане з x. Ці вимірювання моделюються як умовна імовірність $P(z|x)$.

Потім слід обчислити нові імовірності, пов'язані зі станом x з апріорної інформації $P(x)$ та отриманої інформації проведеним вимірюванням [12].

Така модель працює найкраще, якщо розподіли випадкових величин відомі (для багатьох процесів розподіли були виявлені або можуть бути оцінені емпірично з наявних даних за допомогою статистичних тестів). Тоді уточнення апріорних імовірностей є суто аналітичним і формалізованим процесом. Теорема Баєса також легко узагальнюється для більшої кількості випадкових величин, що дозволяє використовувати її в багато сенсорних системах.

Таким чином, у імовірнісних підходах певним чином обирається апріорний розподіл невідомих значень, який коригується шляхом використання інформації інших модальностей та/або вимірювань. Як значення, відновлене таким підходом обирається найбільш імовірне після виконання всіх ітерацій корекції розподілу.

Таким чином, імовірнісну модель можна використовувати для прогнозування стану системи за частково відомими модальностями, вважаючи невідомі значення відсутніми даними та обчислюючи їх найімовірніші значення виходячи з відомих модальностей та спостережень.

Окрім цього, імовірнісні підходи дозволяють обчислити довірчий інтервал отриманого значення, тобто окіл, в якому з попередньо обраною імовірністю знаходиться шукане значення.

1.3.4. *Методи машинного навчання*

Машинне навчання – це область, яка швидко розвивається і знаходить своє успішне застосування в багатьох сучасних задах, в тому числі і поєднанні даних. За допомогою сукупності методів оптимізації методами машинного навчання можна досягти значних успіхів у прогнозуванні та моделюванні систем.

Серед методів машинного навчання, які часто і ефективно застосовуються в задачах аналізу, моделювання та прогнозування слід виділити штучні нейронні мережі та генетичні алгоритми [16].

Штучна нейронна мережа представляє собою множину пов’язаних синапсами штучних нейронів – простих програмних або апаратних сигнальних процесорів. Таким чином, нейрон приймає множину вхідних даних – сигналів – виконує відносно просте обчислення і видає вихідний сигнал.

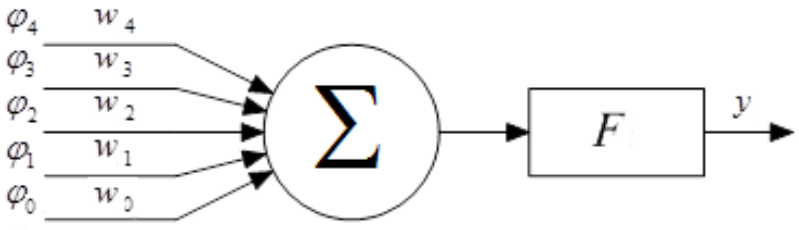


Рис. 1.3. Структура штучного нейрона

На рис. 1.3 зображено загальну структуру нейрона, де φ_i – вхідні сигнали, φ_0 – зміщення (необов'язковий елемент, який завжди видає однаковий сигнал, як правило одиницю), w_i – синаптична вага і-того синапсу, коефіцієнт, який визначає, наскільки сильно даний сигнал впливає на вихідне значення нейрона, $\sum$ – суматор нейрона, який обчислює скалярний добуток векторів вхідних значень і ваг синапсів, отримуючи проміжне значення, яке передається в функцію активації *F*, яка обчислює вихідне значення нейрона *y*, яке іноді називається збудженням.

Нейронні мережі складаються із великої кількості нейронів, які умовно поєднуються у шари. Як зазначалось раніше, нейрони пов'язані між собою синапсами – зв'язками, які характеризуються синаптичною вагою – мірою значимості сигналу, що надходить за даним синапсом. Сам же нейрон обчислює середнє зважене сигналів, що надходять до нього.

Поведінка мережі визначається її топологією та синаптичними вагами, оскільки нейрони завжди виконують одне і те саме обчислення. Тому для вирішення практичних задач необхідно обрати топологію мережі та отримати необхідні синаптичні ваги – навчити мережу.

Навчання мережі – це ітераційний процес, під час якого за допомогою певного метода оптимізації (як правило використовується метод градієнтного спуску) за набором прикладів з відомими вихідними результатами ваги мережі змінюються, щоб якомога більш точно адаптуватись для правильного аналізу тестових прикладів (розглядається варіант так званого навчання з учителем, при якому правильні результати обробки вхідних даних відомі, як в сценарії з певним існуючим набором даних; в наслідок розвитку машинного навчання методи навчання мереж та їх особливості сильно відрізняються в залежності від класу мережі) [17].

Нейронні мережі знайшли широке застосування в ряді задач класифікації, кластеризації та розпізнавання шаблонів, що дозволяє застосувати їх в задачах поєднання даних, які за своєю суттю і є задачам

виявлення та/або використання закономірностей у наборах мультимодальних даних.

Використання штучних нейронних мереж для аналізу мультимодальних даних найчастіше приймає один з двох варіантів: всі модальності утворюють один вхідний вектор (як правило, за допомогою конкатенації, але можливі і інші перетворення для отримання єдиного вектора), який аналізує мережа або навпаки, модальності аналізуються окремо.

Перший підхід більш ефективний у випадку, коли дані мають однакову природу, наприклад виміри датчиків середовища. Тоді все процес розглядається як часовий ряд, точки якого утворюють поєднані в єдиний вектор модальності. Оскільки нейрони пов’язані синапсами з усіма нейронами наступного шару, це дозволяє виявити закономірності у зв’язках між модальностями, а наявність зміщення та нелінійні функції активації дозволяють моделювати складні залежності та вплив шуму на вимірювання. Нейронні мережі ефективно використовуються для аналізу часових рядів, але їх використання для аналізу мультимодальних даних дещо обмежено фактом того, що вхідний вектор є великим, вибір кількості та розмірності прихованих шарів стає дуже важливим фактором, навіть більше ніж зазвичай в побудові нейронних мереж. Оскільки підбір конфігурації мережі виконується експериментально, то кількість спроб необхідних для підбору оптимальних гіперпараметрів та навчання мережі, час та обладнання, необхідні для їх проведення стають неприйнятними.

Інший підхід – використання модальностей розділено з обробкою за допомогою різних мереж. Цей підхід ефективний у випадку, коли набір даних складається з різних за своє природою даних і кількість модальностей відносно невелика, наприклад, набір містить зображення, певний текст, пов’язаний з ним та деякі числові дані (хороший приклад такого набору – пости в соціальних мережах, або результати парсингу сайтів, які містять мультимедійний контент) [18].

В цьому підході модальності обробляються окремо з використанням типів мереж, найбільш ефективних для їх аналізу. Так, у розглянутому прикладі аналіз зображення виконується за допомогою згорткової мережі, для аналізу тексту, як правило, використовують рекурентні мережі, а числові дані обробляються за допомогою персептрона. Після цього, результати роботи кожної задіяної мережі певним чином поєднуються між собою. Найпростіший метод такого поєднання – середнє зважене результатів. Більш складний варіант передбачає використання ще однієї, простої за своє структурою, нейронної мережі для поєднання результатів.

Такий підхід дозволяє розділити задачу аналізу мультимодальних даних на окремі блоки: аналіз зображень, текстів, числових даних, поєднання результатів та розв’язувати їх окремо, при чому допускається часткова заміна або покращення окремих блоків. Однак, даний метод частіше використовується не для моделювання систем, а для отримання надбудови над набором даних. Прикладом його використання є автоматичні системи, які генерують текстовий опис зображення або його ключових характеристик.

Генетичний алгоритм є частиною еволюційних обчислень, що імітують еволюцію на цифровому комп’ютері. За рахунок такого моделювання виникають певні алгоритми оптимізації. Вони базуються на простих правилах. Мета полягає в тому, щоб отримати допустиме рішення задачі оптимізації. Еволюційні обчислення імітують біологічну еволюцію і застосовує імовірнісні оператори пошуку в певному просторі для отримання рішення. Ґрунтуючись на біологічних системах, ГА приймають правила природного відбору та генетики для отримання підходящих рішень.

Так, алгоритм починає роботу з випадковим набором можливих рішень – популяцією – і використовує ітераційний алгоритм еволюції. Спочатку відбувається розмноження, тобто отримання нових рішень за допомогою поєднання існуючих. Як правило, для утворення нового

рішення використовують два існуючих, проте їх кількість може бути довільною, які випадково обираються з існуючої популяції (існують модифікації, в яких випадково обирається тільки один з предків, а вибір інших відбувається на основі його характеристик). В процесі розмноження відбуваються мутації – випадкові зміни в рішенням, які вносять різноманіття в популяцію. Потім, на основі певного критерію пристосованості відбувається відбір найбільш підходящих рішень для наступної ітерації, а найменш підходящі рішення вилучаються. Процес повторюється доки не буде знайдено підходяще рішення, не відбудеться виродження популяції (збіжність до локального оптимуму), або не буде досягнуто максимальної кількості ітерацій (відсутність збіжності за певну кількість ітерацій) [12].

Проблема генетичного алгоритму, як і багатьох методів машинного навчання, полягає в великій кількості параметрів алгоритму, які сильно впливають на його результати і визначення яких потребує проведення експериментів: початкова популяція, метод поєднання рішень, метод внесення мутацій, оцінка пристосованості, тощо. Також, генетичні алгоритми мають тенденцію до збіжності до локального оптимуму або взагалі довільної точки, що робить їх використання ненадійним.

1.3.5. *Основні недоліки розглянутих методів*

Розглянемо недоліки існуючих методів, які потребують покращення і розроблення нових методів поєднання даних. При чому розглядатимемо в основному методи факторизації, оскільки імовірнісні підходи та методи машинного навчання є доволі специфічними, в основному через свою недетерміновану природу та нечіткі результати і їх особливостям, недолікам, перевагам та методам покращення можна цілком присвятити окрему роботу.

Основною проблемою методів факторизації, окрім пошуку змістовних обмежень, які зумовлюють унікальність розкладення з

точністю до перестановки є формування єдиної лінійної моделі для системи, і хоча такий підхід є допустимим і в деяких випадках дає задовільні результати, дослідження і аналізу потребують системи, що генерують велику кількість гетерогенних даних. Такі системи складні і рідко точно апроксимуються однією лінійною моделлю, навіть якщо вдається домогтись унікальності розкладення.

Таким чином, предметом даної роботи є метод, що дозволить більш точно моделювати складні системи, які проявляють різну поведінку в декількох станах, що не вдається добре апроксимувати однією лінійною моделлю.

1.4. Висновки до розділу I

У даному розділі було розглянуто проблему поєднання даних для систем, об’єктів, процесів або явищ, які генерують великий обсяг мультимодальних даних, а саме необхідність створення методу для побудови цифрового двійника системи, який дозволяє моделювати її та виявляти аномалії в певних точках даних за заданим повним та узгодженим набором спостережень.

Було розглянуто існуючі методи поєднання даних: методи матричної та тензорної факторизації, імовірнісні підходи та методи машинного навчання, обговорено їх суть, переваги та недоліки, а саме, для методів машинного навчання – складність підбору параметрів методу, проведення і особливо верифікація навчання моделі. Навіть якщо модель показує прийнятні результати на тестових даних, будь-який її результат не можна прийняти як достовірний, оскільки процес його отримання не є послідовним алгоритмом, який можна легко проаналізувати та довести коректність, і хоча методи верифікації моделей машинного навчання, а особливо штучних нейронних мереж, активно розвиваються, на даний момент їх ефективної реалізації немає.

Подібний недолік має і імовірнісний підхід – його результати базуються на певному розподілі спостережень, але результати прогнозування та моделювання є оцінками величин і природно мають певне відхилення, розуміти і контролювати яке може бути не просто.

Методи факторизації є більш чіткими, але пропонують використання однієї лінійної моделі для всієї системи, яка може бути складною та демонструвати різну поведінку. Тому необхідно створити метод, який дозволить врахувати цю особливість і підвищити точність моделі.

2. СТВОРЕННЯ МЕТОДУ ПОЄДНАННЯ МУЛЬТИМОДАЛЬНИХ ДАНИХ

Враховуючи недоліки існуючих методів, необхідно створити метод, який враховує нелінійність систем та різну їх поведінку в певних станах. Оскільки предметом дослідження є метод поєднання даних, то також потрібно отримати більше інформації для моделювання, ніж можна отримати, аналізуючи модальності окремо.

2.1. Абстрактний алгоритм методу

Враховуючи абстракту природу вхідного набору даних, розглянемо узагальнений метод побудови цифрової моделі. Алгоритм реалізації даного методу потребує уточнення для кожного окремого випадку, щоб врахувати особливості задачі, що розв’язується.

Основна його ідея полягає в застосуванні кластеризації для виявлення закономірностей в даних, які можна інтерпретувати як стани або як підпростори, в яких виконуються певні передумови, наприклад лінійна залежність в даних, що дозволяє апроксимувати дані з більшою точністю.

В загальному випадку, метод складається з таких етапів:

1. Підготовка набору даних.
2. Кластеризація даних.
3. Апроксимація на кластерах.
4. Аналіз та використання отриманої моделі.

Виділення необхідних закономірностей має здійснюватись не тільки за значеннями даних, оскільки навіть розкидані у просторі дані можуть достатньо точно апроксимуватись одним лінійним наближенням, при чому кластеризація лише за значеннями даних дає чіткий поділ (рис. 2.1). Приходимо до висновку, що окрім безпосередньо значень точок даних, кластеризація повинна враховувати і їх статистичні співвідношення або інші особливості, в залежності від поставленої мети поділу простору.

Надалі загальним припущенням буде відсутність попередніх уявлень про закономірності в основі системи, що моделюється, тому ми зосередимось на чисельній апроксимації лінійними наближеннями.

Як приклад додаткового параметра, що дозволяє враховувати співвідношення даних виступає коефіцієнт кореляції, який і показує ступінь лінійного зв'язку між точками даних. Коваріація та кореляція корисні для розріджених даних, оскільки в них такі залежності проявляються більш виражено, в той час як для наборів з високою густиною кореляція між даними у великому околі кожної точки (якщо не у всьому наборі) дорівнює 1 для всіх точок в ньому, що не є показовою або корисною інформацією.

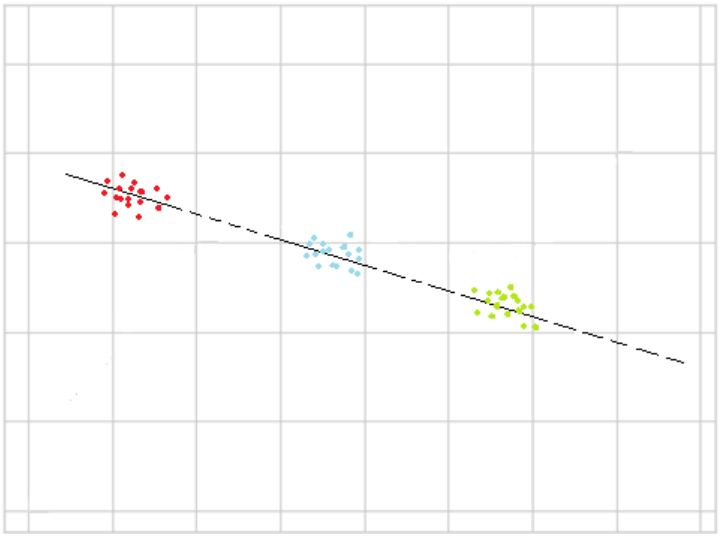


Рис. 2.1. Одне наближення для окремих груп точок

2.2. Аналіз етапів методу

Розглянемо детальніше етапи методу, їх призначення та можливі реалізації. Для кожної конкретної задачі, реалізації етапів їх послідовність та параметри обираються з урахуванням необхідних операцій, досвіду дослідника, встановлених практик та результатів експериментів.

Відповідно, можлива велика кількість інших можливих реалізацій та узагальнень методу, особливо для підготовки даних, оскільки цей етап є

найбільш загальним для всіх задач аналізу даних і визначається багатьма факторами.

2.2.1. *Попередня обробка набору даних*

Підготовка набору даних з програмної точки зору необхідна для приведення вхідного набору в форму, прийнятну для наступних кроків, наприклад, ініціалізація структур даних, приведення типів, тощо.

Окрім цього, підготовка набору даних включає додаткові дії над набором даних, наприклад:

- Відновлення відсутніх значень або вилучення неповних кортежів.
- Очистка даних.
- Попередня обробка набору даних.
- Групування та видалення тотожних кортежів.
- Фільтрація.
- Скорочення даних.
- Доповнення даних.

Хоча в задачах поєднання даних є важливим використання додаткової інформації із різних модальностей, далеко не всі вони мають значимий або взагалі будь-який вплив на інші, тому на етапі підготовки слід обрати незалежні величини, які необхідно прогнозувати за допомогою моделі та відповідні їм набори залежних величин. Визначити зв’язок між модальностями можна за допомогою кореляційного аналізу та візуалізації, при чому окрім кореляції Пірсона варто застосувати інші характеристики, що є менш чутливими до відхилень, наприклад, кореляція Спірмена. Таким чином, можна отримати інформацію про зв’язки між модальностями, навіть якщо

В задачах аналізу даних підготовка є важливим етапом на якому базуються всі подальші кроки та отримані висновки [19, 20].

Відновлення значень

Для використання наступних кроків алгоритму набір даних вважається повним, тому відсутні значення в кортежах даних, які пов’язані з різною частотою вимірювання параметрів, відсутніми або пошкодженими даними, необхідно відновити відповідними методами апроксимації або вилучити їх з набору даних.

Методи відновлення значень є окремою темою досліджень, в тому числі результати поєднання даних часто використовуються з цією метою [21, 22]. Для цього можна застосувати описані раніше методи поєднання мультимодальних даних, в тому числі, і модель, отриману в результаті роботи даного методу, щоб отримати певне наближення значень відсутніх модальностей.

В інакшому випадку можна просто вилучити їх із розгляду, щоб не вносити додаткових похибок при подальшому моделюванні.

Попередня обробка набору даних

Також у випадку мультимодальних даних часто виникає проблема представлення, що випливає з їх різноманітної природи. Перед застосуванням методів поєднання даних, необхідно узгодити їх подання в наборі. Наприклад, бажано привести значення параметрів до однієї системи одиниць виміру, але цей крок не є завжди обов’язковим, оскільки деякі величини краще підлягають розумінню в певному масштабі. Так, наприклад, якщо довжина вимірюється в метрах, а результати вимірювань наведено в міліметрах, то здійснювати таке приведення, як правило, не обов’язково, через те, що по-перше, в предметній області значно легше оперувати значеннями, такими як 3 мм, а не 0.003 м, по друге, таке перетворення може погіршувати обумовленість задачі, по-третє, якщо таке перетворення змістовно важливе, воно буде відображено в коефіцієнтах моделі, що стосуються даної величини.

Окрім приведення до певної розмірності, застосовуються масштабування або певні перетворення, що роблять більш або менш значущим відхилення величин, особливо в порівнянні з іншими модальностями в наборі.

Також слід попередньо проаналізувати види даних, присутні у наборі, оскільки там можуть міститись аналогові та/або дискретні величини, зображення, відео/аудіо записи, тощо.

Для більш ефективної роботи методу можна перетворити набір так, щоб природа даних була більш однорідною, наприклад, ввести кодування для дискретних параметрів, а зображення перетворити на вектори ознак, які містять суттєву з точки зору поставленої задачі інформацію за допомогою алгоритмів виділення ознак або розпізнавання образів або обчислити певні інтегральні характеристики, наприклад інтенсивність або розподіл кольорів на зображенні, щоб перетворити його на вектор чисел, який значно легше обробляти, ніж саме зображення і отримати при цьому не менше корисної інформації.

Групування

Вхідний набір мультимодальних даних також можна очистити від значень, що не містять змістовної інформації. Таким чином, тотожні кортежі можна видалити з набору, оскільки вони не містять додаткових відомостей про систему, а тільки потребують додаткових ресурсів на обробку та збільшують похибку моделювання.

Також, часовий інтервал між вимірюваннями може сильно впливати на характеристики та особливості набору даних. При великій частоті вимірювань отримана інформація також носить характер шуму, особливо у випадку, коли досліджуваний процес або стан системи змінюється повільно (відносно частоти вимірювання параметрів), наприклад сенсори інтернету речей можуть генерувати величезну кількість даних з високою частотою, які містять велику кількість тотожних або дуже близьких кортежів. В такому випадку можна обрати часові проміжки, в яких

відхилення параметрів є відносно малим (метрику відхилення та значення порогу можна обрати на розсуд дослідника, наприклад, величиною, яка описує зміну значень може бути евклідова норма різниці кортежів) і згрупувати дані в цьому проміжку і обчисливши в ньому середнє значення, значно зменшивши кількість даних для обробки. Вважаючи, що система інваріантна в часі, це призводить до суттєвого зменшення об’єму даних без втрати важливої інформації.

Так само, можна згрупувати між собою кортежі з невеликим відхиленням значень отриманих не внаслідок частих вимірювань параметрів, а в силу поведінки системи, за умови, що час не є важливим параметром, що аналізується. Таким чином, для кортежів, у яких норма різниці не перевищує деякої невеликої наперед заданої величини можна обчислити середнє значення щоб зменшити обсяг даних.

Фільтрація

Якщо дані надходять безпосередньо з сенсорів у сирому вигляді, вони містять шум, викликаний будовою самих сенсорів та систем моніторингу, середовищем, в якому функціонує система та проводяться вимірювання та іншими факторами.

Для зменшення впливу шуму на точність моделювання системи до вимірів застосовують алгоритми цифрової фільтрації, які певним чином трансформують вхідний сигнал, щоб отримати бажані характеристики.

Наприклад, для згладжування сигналів в часовій області використовуються такі фільтри, як рухоме середнє та інтегратор з втратами, і дозволяють отримати значення даних, які більше відповідають динаміці змін параметрів системи.

Для виділення шуму з сигналу застосовується частотний аналіз, який розглядає спектр сигналу в частотній області і за допомогою фільтрації низьких або високих частот дозволяє ефективно виділити змістовну інформацію із зашумленого сигналу [23].

Скорочення даних

У випадку дуже великого за об’ємом набору даних, який складно аналізувати звичайними методами, до нього можна застосувати методи скорочення даних, які широко досліджуються в big data.

Скорочення даних – це перетворення цифрової інформації у скориговану, упорядковану та спрощену форму. Мета зменшення даних може бути подвійною: зменшити кількість записів даних, усуваючи недійсні дані, або створювати зведені дані та статистику на різних рівнях агрегування для подальшого застосування аналітичних методів. Коли інформація отримується з показань приладів, може також відбуватися перехід від аналогової до цифрової форми. Коли дані вже перебувають у цифровій формі, скорочення даних, як правило, передбачає певне редагування, масштабування, кодування. Коли спостереження дискретні, але основне явище неперервне, тоді часто потрібні згладжування та інтерполяція [24, 25].

Методи скорочення даних поділяються на скорочення розмірності простору та скорочення кількості даних. В контексті аналізу мультимодальних даних скорочення розмірності простору, як правило, не застосовується, хоча деякі перетворення можна виконати, не втрачаючи суттєвої інформації, наприклад, аналіз головних компонент може бути застосований для скорочення розмірності простору на етапі підготовки, а не як метод розв’язання задачі поєднання даних.

Коли розмірність простору зростає, дані стають розрідженими, тоді як щільність та відстань між точками, що є критично важливими для кластеризації та аналізу, стають менш значущими. Зменшення розмірності допомагає зменшити шум у даних і полегшує візуалізацію.

Методи кількісного зменшення даних зменшують обсяг даних, вибираючи компактніші форми подання. Зменшення кількості можна розділити на дві групи: параметричні та непараметричні методи. Параметричні методи передбачають, що дані відповідають деякій моделі,

оцінюють параметри моделі, зберігають лише параметри та відкидають дані. Враховуючи, що дане дослідження присвячене пошуку моделей, що адекватно апроксимують набір даних, параметричні методи не можуть бути ефективно використані в даному методі.

Непараметричні методи не передбачають моделювання, наприклад, вибірка, або групування, яке обговорювалось раніше.

2.2.2. *Кластеризація даних*

Як було показано, системи мають різні стани та відповідну поведінку в них, яка у випадку складних систем та/або великої кількості вимірюваних параметрів має нелінійний характер. Щоб врахувати таку поведінку, можна застосувати розподілення точок даних, яке дозволяє виділити їх взаємозв'язок. Враховуючи неповні або взагалі відсутні відомості про функціонування системи, залишається виявляти закономірності в даних, як носії інформації та її відображення.

Досягти цього можна за допомогою методів виявлення невідомих закономірностей в даних, а саме кластеризації.

Не зважаючи на свої обмеження, лінійні наближення мають ряд переваг, через які вони часто застосовуються, а саме ефективні та досліджені методи визначення параметрів, обчислювально просте застосування отриманих моделей, параметри аналізу та оцінки. Тому поставимо задачу виділення підмножин простору значень, в яких лінійні наближення будуть достатньо точними.

Слід зауважити, що на даному етапі сильний вплив мають попередні припущення про процеси, які лежать в основі системи, що досліджується. Вважаючи що вони відсутні, ми розглядаємо лінійні моделі, параметри яких можна швидко обчислити за допомогою алгоритмів лінійної регресії і які чисельно моделюють систему.

Якщо існує припущення, що залежність між модальностями підпорядковується деякому набору (особливо нелінійних) законів,

кластеризація повинна враховувати ці припущення і виконувати поділ простору, за якого моделі цих законів будуть точними. Вибір лінійних наближень в цьому сенсі зумовлений наявністю характеристик, що їх виявляють, таких як коваріація та кореляція, в той час як для довільних нелінійних залежностей складно застосувати прості метрики для оцінки їх відповідності даним і, так само як і для даних з високою густиною при застосуванні лінійних наближеннях залишається обчислювати параметри моделі в деякому околі початкової точки та формувати кластер навколо даного околу, після чого повторити процес для інших даних, що не увійшли в даних кластер.

Щоб наочно показати суть даного етапу, розглянемо простий приклад двовимірного нелінійного процесу, вважаючи, що закон, що лежить в його основі не відомий (рис. 2.2).

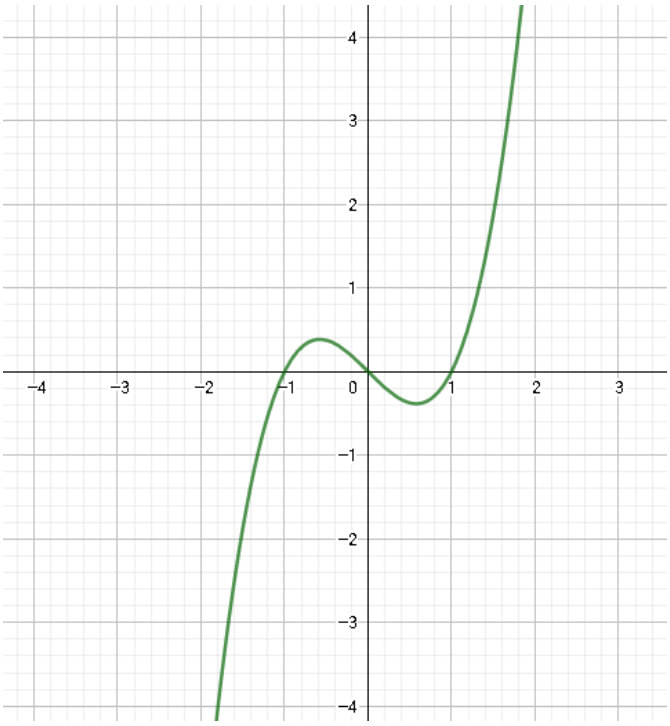


Рис. 2.2. Нелінійна залежність

Як набір даних для аналізу розглянемо вибірку точок з даного процесу з невеликим зашумленням (рис. 2.3). Дана вибірка сильно

відрізняється від реальних наборів мультимодальних даних, але з її допомогою можна дослідити важливі аспекти даного етапу, які достатньо просто узагальнюються на складніші набори з більшою розмірністю.

Застосування одного лінійного наближення до даного набору даних призведе до неточної апроксимації в силу нелінійності процесу, що розглядається. Методи, розглянуті раніше, такі як аналіз головних компонент можуть дати більш точне наближення, але за рахунок збільшення кількості ступенів свободи системи, яким потрібно надати змістовну інтерпретацію.

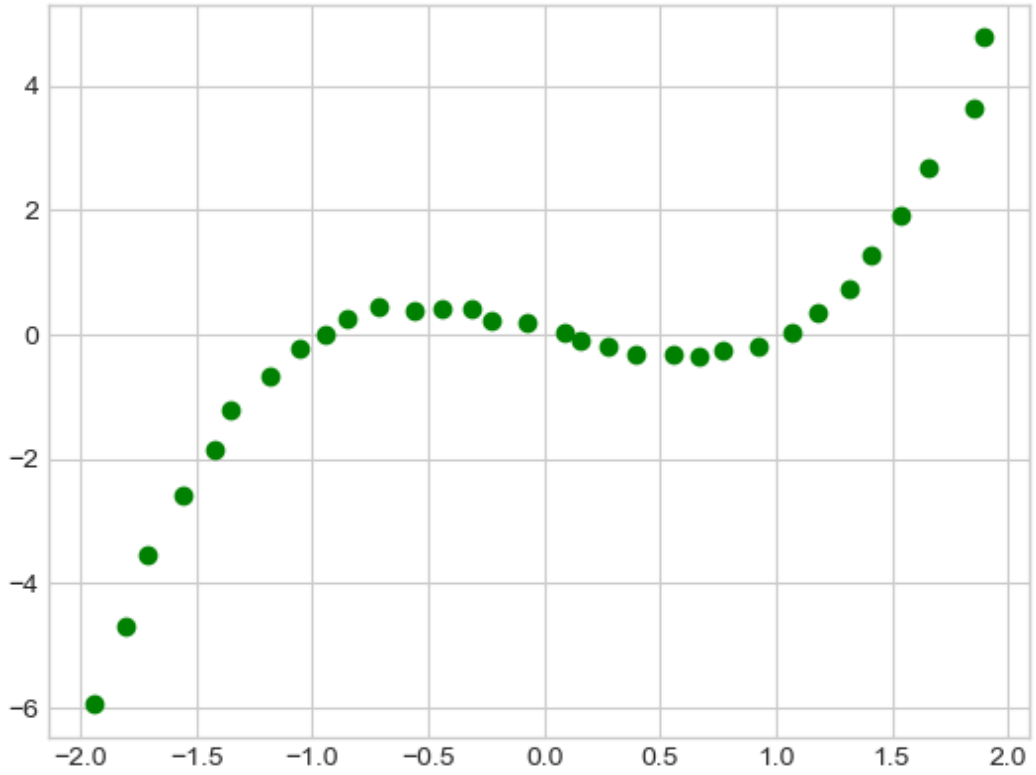


Рис. 2.3. Вибірка значень процесу

Замість цього пропонується розділити набір даних на фрагменти, в яких прості лінійні наближення будуть достатньо ефективними. Враховуючи поставлену задачу, і як наголошувалось раніше, застосування методів кластеризації безпосередньо на значеннях точок не дає бажаного результату, оскільки вони визначають закономірності в самих значеннях даних, а не їх співвідношеннях (рис. 2.4), завдяки чому можна впевнитись,

що кластеризацію потрібно проводити не лише за самими значеннями точок даних, а за статистичними співвідношеннями між окремими точками та модальностями вцілому, які свідчать про певний взаємозв'язок, який буде моделюватись на наступному етапі.

Є декілька варіантів виявлення частин простору, в яких поведінка системи близька до лінійної. Мірою лінійної залежності величин є коефіцієнти коваріації та кореляції. Застосувавши їх для статистичного аналізу даних, можна виявити шукані закономірності. Для даних з високою густиною кореляція втрачає ефективність в відображенні лінійних залежностей, тому пропонується ітеративна побудова кластерів на основі обчислення параметрів моделі в околі початкової точки та розширення кластера допоки наближення не втратить точності або сильно не відхилиться від попереднього отриманого таким чином. За логікою такий алгоритм схожий на алгоритм DBSCAN, в якому функція густини враховує параметри наближення в кластері.

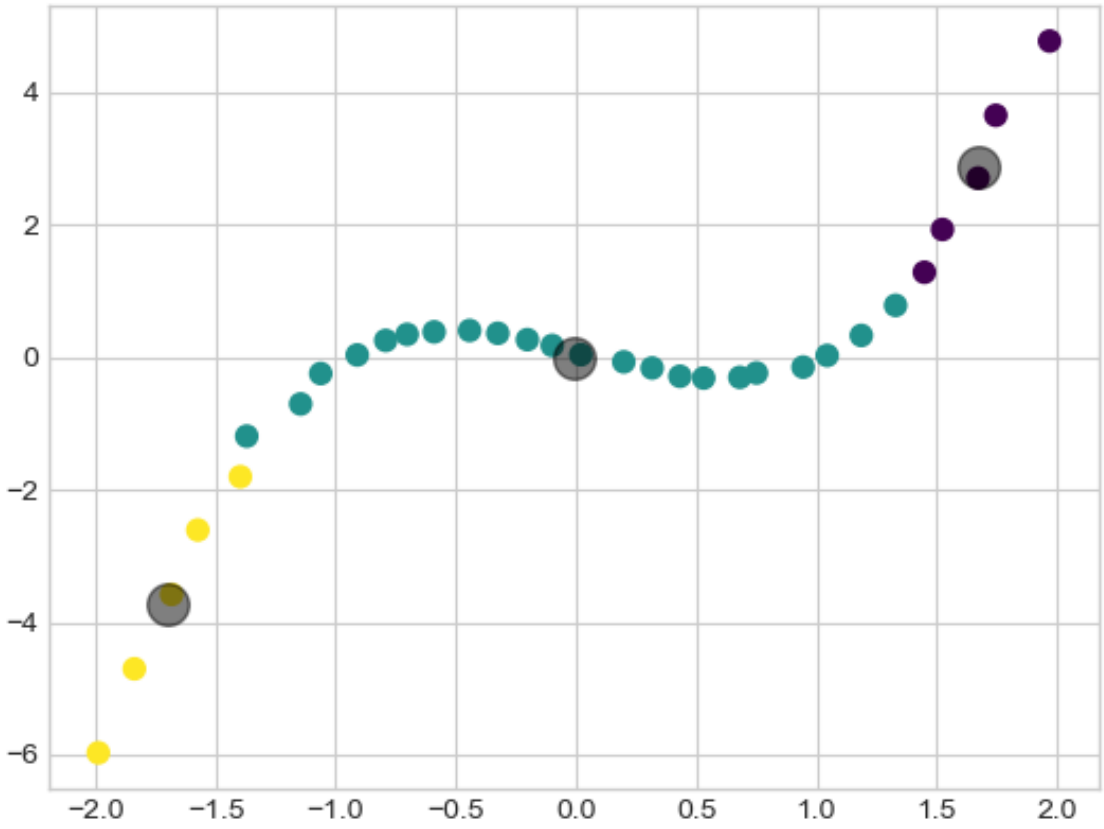


Рис. 2.4. Кластеризація k-means набору даних

Наприклад, для розглянутого прикладу процесу, використання кореляції Пірсона призводить до наступного результату кластеризації, наведеному на рис. 2.5.

Зі збільшенням розмірності задачі і використанні більшої кількості модальностей, складність методу зростає через необхідність аналізу матриць кореляції для модальностей, але суть даного етапу не змінюється і полягає в пошуку кластерів даних, які достатньо точно моделюються лінійними наближеннями.

В наведеному прикладі дані достатньо розріджені і використання коефіцієнту кореляції дозволяє достатньо точно виділити залежності, близькі до лінійних. В реальних задачах дані мають значно більшу густину і навіть в інваріантих у часі системах, які розглядаються в даній роботі мають нефункціональну залежність, тобто одному набору значень незалежних величин може відповідати декілька значень залежної, що призводить до значного ускладнення: приналежність точки кластеру визначається як значеннями залежних величин, так і значенням незалежної.

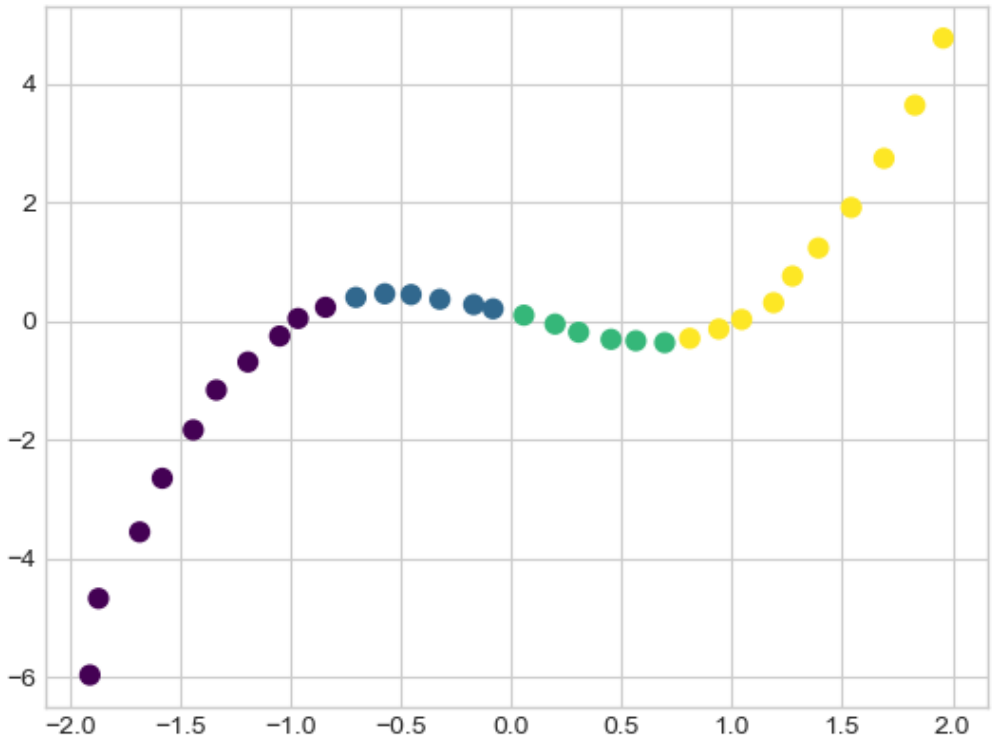


Рис. 2.5. Кластеризація даних з урахуванням коефіцієнта кореляції

Це означає, що під час використання моделі, тобто оцінці залежних величин за незалежними, неможливо однозначно визначити приналежність точки до кластеру лише за значеннями незалежних величин. На даному етапі для розв'язання цієї проблеми використаємо поділ простору (гіпер)площинами, що паралельні осям незалежних величин і рівновіддалені від центрів кластерів, що дозволяє визначити кластер однозначно, однак в подальшому метод поділу простору на інтервали, які однозначно визначаються значеннями незалежних величин може бути іншим.

2.2.3. *Апроксимація на кластерах*

Після виділення інтервалів за статистичними характеристиками даних, для кожного з них застосовується моделювання, використовуючи дані, які відносяться до даного кластеру.

Моделі, що будуються на даному етапі відповідають тим припущенням, які зумовлюють отриманий поділ простору значень, оскільки він був отриманий для досягнення більшої точності з використанням таких моделей.

Так, за відсутності апріорних уявлень про функціонування системи, кластери на попередньому етапі формувались таким чином, щоб забезпечити більш точну лінійну апроксимацію, тому її використання є виправданим рішенням.

Лінійні моделі, що будуються на цьому етапі можна отримати за допомогою методів оптимізації, прийнятних для наявного обсягу та характеру даних.

В загальному випадку для модальностей x_1, x_2, …, x_n, що корелюють на даному кластері будується співвідношення $x_k = \sum_{i=1, i\neq k}^{n} a_i x_i + a_0$ або еквівалентне йому $a_0 + \sum_{i=1}^{n} a_i x_i = 0$, коефіцієнти якого a_i, $i = \overline{0..n}$ обчислюються за методом найменших квадратів, тобто обчислюється

оцінка параметрів шляхом мінімізації квадратів відхилення зазначеної вище суми від нуля для всіх точок, що належать інтервалу.

У випадку великої кількості таких точок та модальностей обчислення коефіцієнтів за методом найменших квадратів стає обчислювально витратним і слід скористатись іншими оптимізаційними методами для обчислення параметрів регресійної моделі, наприклад, метод градієнтного спуску, який широко використовується для в оптимізаційних задачах машинного навчання.

Так, наприклад, для двовимірного прикладу, розглянутого раніше використання лінійних наближень на виділених кластерах призводить до результату, зображеному на рис. 2.6.

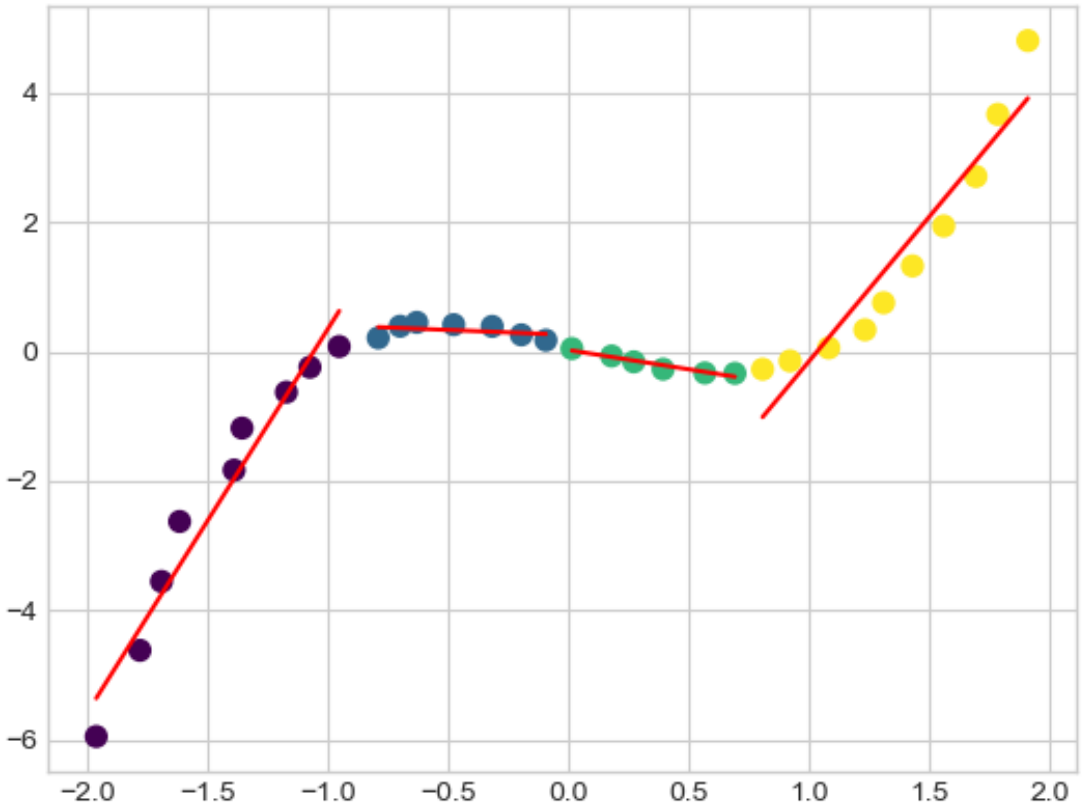


Рис. 2.6. Використання лінійних наближень на кластерах

Отримані моделі дають точніше наближення до вихідних значень, ніж використання одного наближення для всього набору даних. Втім, їх оцінка показує відхилення від точних значень. Найбільше відхилення

спостерігається на межі кластерів, що пов’язано з особливостями процесу та методу обчислення параметрів регресійних моделей.

Раніше обговорювались можливість створення моделей для декількох незалежних величин. В цьому контексті застосування методу передбачається для кожної незалежної величини окремо, оскільки вони визначають різні інтервали моделювання.

Кластеризація та моделювання не є чіткими визначеними алгоритмами і мають адаптуватись для кожної окремої задачі, а вибір найбільш ефективних для цього підходів здійснюється експериментально за допомогою аналізу результатів, тому для досягнення необхідної точності може бути потрібно проведення декількох спроб моделювання з різними параметрами і вибір з них найкращої.

2.2.4. *Аналіз отриманої моделі*

Після проведення кластеризації та моделювання потрібно проаналізувати отриману модель системи.

Першим кроком є оцінка точності за наявними даними (верифікація). Для цього невелику кількість вхідних даних (10-15% загальної кількості) слід залишити як тестові і не використовувати під час моделювання. Перевірка з використанням цих даних дозволяє більш якісно оцінити точність створеної моделі. Якщо точність на тестових даних задовільна, можна продовжити аналіз моделі, інакше слід повторно застосувати метод з іншими параметрами. За необхідності можна також застосувати статистичний аналіз і виявити імовірнісні характеристики точності, такі як довірчі інтервали для значень, що моделюються з деякою заданою наперед імовірністю.

Метрикою, за якою здійснюється верифікація як правило виступає середнє квадратичне відхилення, але в контексті даної задачі корисним показником також є середнє абсолютне відхилення, яке є показником

нев’язки моделі на тестових даних і в подальшому буде використовуватись для виявлення аномалій в даних [26].

Потрібно також детальніше розглянути виділені інтервали та моделі. Їх кількість може бути великою у випадку великої кількості модальностей та кортежів у наборі даних, при цьому якщо вони знаходяться поруч у просторі та мають близькі параметри моделей, то їх можна поєднати в один, оскільки вони відповідають однаковій поведінці системи.

Верифікація також дозволяє виявити інтервали, на яких точність є задовільною. Оскільки моделювання відбувається в частині простору, в якій наявні дані, складно судити про валіднісь створеної моделі поза нею та на її межах. Так само може виявитись, що точність нижча в окремих кластерах, що дозволить зменшити кількість обчислень при повторному моделюванні, обмежившись лише тими інтервалами, де точність не є задовільною.

Після цього виділеним кластерам та моделям слід надати певного фізичного сенсу. В найкращому випадку кластери відповідають різним станам системи, в яких її поведінка відрізняється, інакше маємо просто чисельне наближення, яке описує систему.

На основі отриманих параметрів моделювання можна побудувати уявлення про функціонування системи та зв’язки між параметрами, що спостерігаються, особливо якщо можна зручно візуалізувати отримані результати.

2.3. Застосування створеної моделі для оцінки значень параметрів та виявлення аномалій

Після створення та верифікації моделі, її завдання полягає в виявленні аномалій в станах системи та відновлення значень модальностей, інформованих іншими вимірами.

Для виявлення аномалій в кортежі даних необхідно знайти інтервал, до якого належить дана точка даних. Для моделі, створеної для даного

кластеру (вважаючи їх вигляд $a_0 + \sum_{i=0}^{m} a_i x_{k_i} = 0$ або в загальному вигляді $f(\bar{x}) = 0$) обчислюються абсолютна величина нев'язки $\delta = |f(\overline{x^*})|$.

Мірою відхилення є відношення отриманої нев'язки до середнього абсолютного відхилення, отриманого під час верифікації моделі. Відношення, що значно перевищують 1 вважаються аномальними, проте точне значення, яке є достатнім для визначення аномалій не є визначеним і залежить від задачі.

Прогнозування здійснюється за допомогою пошуку інтервалу, до якого належить дана точка значень залежних величин та застосування відповідної моделі цього інтервалу для отримання значення незалежної величини.

2.4. Висновки до розділу II

У даному розділі було описано абстрактний метод поєднання мультимодальних даних для систем з нелінійною поведінкою, інваріантних в часі (тобто час виміру та його вплив на поведінку системи не є визначним фактором, процеси відбуваються майже однаково і визначаються станом системи).

Було описано призначення та можливі реалізації етапів методу. Він є проміжним між методами керованим даними, та методами, керованими моделями, оскільки хоча його суть полягає в пошуку інтервалів, для яких будуть ефективними певні види наближень, пошук цих інтервалі відбувається на основі залежностей в даних, що виявляються за допомогою кластерзації. За відсутності попередніх уявлень про закономірності в досліджуваній системі, методу передбачає відшукання інтервалів у просторі параметрів, в яких лінійні наближення будуть достатньо точними.

Як і будь-який метод інтелектуального аналізу даних, запропонований метод містить багато невизначених параметрів, які складно апріорно оцінити, тому для їх визначення потребується

проведення декількох експериментів з наявним набором даних і ефективність його сильно відрізняється від задачі до задачі.

Етапи методу передбачають використання багатьох суміжних дисциплін, таких як цифрова обробка сигналів, статистика, big data, машинне навчання, тощо для оброки та аналізу набору даних.

3. ПРОГРАМНА РЕАЛІЗАЦІЯ МЕТОДУ

Для більш ефективного використання методу та проведення експериментів для визначення параметрів, необхідно створити програмне забезпечення, яке ефективного його реалізує, дозволяє гнучке розширення додатковими реалізаціями етапів методу та надає засоби візуалізації та аналізу результатів.

Основним призначенням розроблюваного ПЗ є дослідження запропонованого методу, його особливостей та ефективності, більше ніж безпосереднє моделювання систем. Тому воно орієнтовано на користувача, добре ознайомленого з будовою методу та його проміжними результатами і має надавати засоби їх (проміжних результатів) дослідження, візуалізації та оцінки за такими критеріями, як точність моделювання, кількість інтервалів після розбиття та відповідно, побудованих моделей, обсяг пам’яті що використовується для їх зберігання.

Створення ПЗ відбувається відповідно до звичного життєвого циклу: виявлення вимог, проєктування, реалізація, тестування, використання, супровід, виведення з експлуатації.

3.1. Вимоги до розроблюваного програмного забезпечення

Визначимо вимоги, яким має відповідати програмне забезпечення, що розробляється.

Його функціональність пов’язана з реалізацією етапів методу та зборі проміжної інформації про їх результати та подальшого дослідження та покращення, що відображається на діаграмі варіантів використання (рис. 3.1).

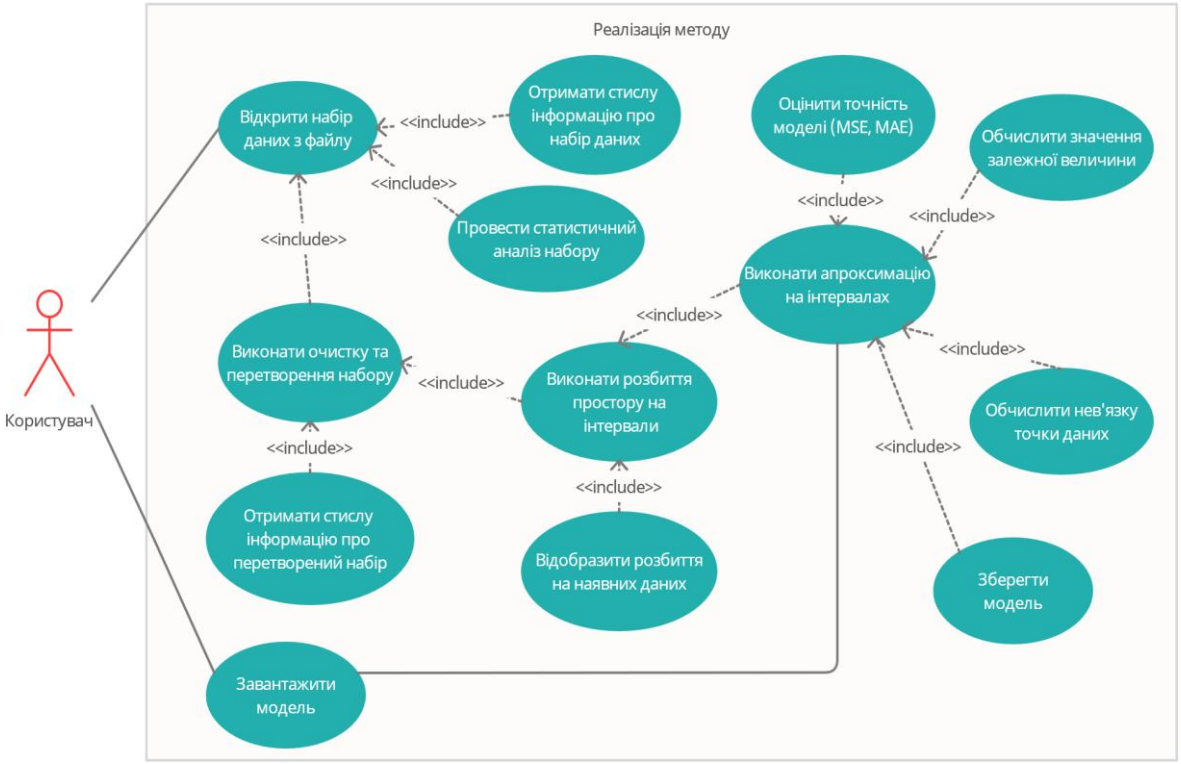


Рис. 3.1. Діаграма варіантів використання ПЗ

Більш детальна специфікація та пояснення наводяться в реєстрі вимог (табл. 3.1).

Таблиця 3.1

Реєстр вимог

№	Вимога	Опис
Функціональні вимоги		
1	ПЗ має реалізовувати метод поєднання мультимодальних даних	Розроблюване програмне забезпечення має реалізовувати алгоритм виконання описаного методу агрегації мультимодальних даних
2	Введення набору	Програмне забезпечення приймає на вхід набір мультимодальних даних у форматі csv, xml або json.
3	Перевірка набору	Необхідно виконати перевірку зчитаного набору даних на відповідність вимогам методу та відобразити короткі відомості про набір: модальності, виявлені типи даних, статистичні характеристики.

Продовження табл. 3.1

4	ПЗ має реалізовувати метод поєднання мультимодальних даних	Розроблюване програмне забезпечення має реалізовувати алгоритм виконання описаного методу агрегації мультимодальних даних
5	Введення набору	Програмне забезпечення приймає на вхід набір мультимодальних даних у форматі csv, xml або json.
6	Перевірка набору	Необхідно виконати перевірку зчитаного набору даних на відповідність вимогам методу та відобразити короткі відомості про набір: модальності, виявлені типи даних, статистичні характеристики.
7	Можливість вибору реалізації етапів	Програмне забезпечення має надавати можливість вибору реалізації етапів методу з набору можливих
8	Аналіз і звітність етапів	Програмне забезпечення дозволяє переглянути та візуалізувати результати етапів методу
9	Збереження та відновлення моделей	Програмне забезпечення має забезпечувати спосіб збереження створених цифрових двійників та їх відновлення зі збереженого стану
10	Візуалізація результатів	Необхідно виконати візуалізацію проміжних та кінцевих результатів роботи методу.
Нефункціональні вимоги		
11	Наявність графічного інтерфейсу користувача	Програмне забезпечення має надавати графічний інтерфейс користувача роботи з даними та візуалізації результатів
12	Стабільність	Програмне забезпечення коректно завершує роботу та звільняє ресурси. Помилки обробляються та відповідні повідомлення виводяться користувачу

Продовження табл. 3.1

13	Наявність керівництва користувача	Програмне забезпечення має супроводжуватись довідковою документацією, яка пояснює його функції та роботу реалізованого методу
Вимоги до процесу розробки		
14	Розширюваність	Структура програмного забезпечення має передбачати розширення новими методами реалізації етапів (бажано, але не обов’язково динамічно)
15	Відповідність стандартам коду обраних технологій	Код програмного забезпечення має відповідати галузевим стандартам мов та технологій, обраних для розроблення

3.2. Архітектура системи

Архітектура програмної системи – це базова високорівнева організація системи, що визначає її основні компоненти та способи взаємодії між ними. Архітектура визначає загальні абстрактні елементи, що забезпечують функціонування структурних блоків системи, і має безпосередній вплив на засоби і принципи розроблення, оскільки обумовлює конкретні проєктні рішення для дотримання спроєктованих архітектурних конструкцій.

Правильно обрана архітектура системи дозволяє організувати процес розроблення та супроводу системи значно легше, особливо у випадках, коли доводиться регулярно модифікувати та розширювати програмні модулі, завдяки ізоляції компонентів та виділенню абстрактних зв’язків між ними, що дозволяє змінювати та впроваджувати а тестувати можливості програмної системи локально, не впливаючи на інші її частини. Такий підхід дуже важливий для сучасного програмного забезпечення, тому було виділено певні архітектурні шаблони – перевірені ефективні рішення дизайну архітектури, які мають певні передумови застосування, переваги та недоліки, найпоширеніші серед яких:

- багаторівнева архітектура;
- мікросервісна архітектура;
- мікроядерна архітектура;
- керована подіями архітектура.

Окрім зазначених шаблонів, існують також їх різноманітні модифікації та узагальнення, використання яких, як правило зумовлено специфікою проєктів, в яких вони застосовуються, тому при виборі архітектурного шаблону вони зазвичай не розглядаються, а імплементуються за необхідності в певних умовах [27].

Система розробляється з використанням шаблону багаторівневої архітектури, який передбачає розділення функціональності системи на окремі незалежні блоки – рівні, які взаємодіють ніж собою через визначені програмні інтерфейси і можуть бути замінені на іншу реалізацію, що реалізує зазначений інтерфейс, при цьому ніяк не впливаючи на інші рівні, що дозволяє створити більш гнучку та простішу у супроводі, модифікації та тестуванні систему. Багаторівнева архітектура є хорошим початковим рішенням для розроблення та тестування програмного забезпечення. Враховуючи особливості програмного забезпечення, що розробляється, також доцільним було б використання мікросервісної архітектури для паралельного використання окремих елементів в декількох інстансах в мережі. Відповідні можливості виокремлення компонентів в сервіси будуть розглянуті під час обговорення результатів роботи. Втім, на етапі початкового проєктування та розроблення використання багаторівневої архітектури дозволяє значно легше виявити структурні компоненти, необхідні зв’язки та інтерфейси їх взаємодії. Також тестування та верифікація роботи компонентів та їх зв’язків простіша з використанням багаторівневої архітектури в порівнянні з мікросервісною.

В практичному застосуванні методу для побудови цифрових двійників при роботі з великими наборами даними також необхідне

застосування архітектурних шаблонів Big data [5], проте на етапі дослідження методу їх впровадження не є виправданим.

Таким чином, система складається з рівнів, зображених на рис. 3.2.

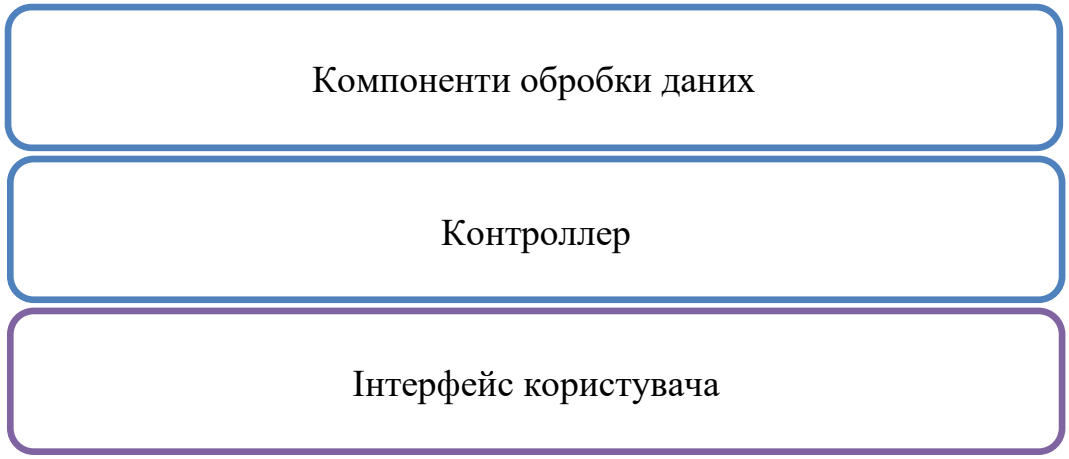


Рис. 3.2. Рівні архітектури системи

Інтерфейс користувача надає зручні засоби вибору наборів даних, реалізацій етапів методу та візуалізації результатів. Задача інтерфейсу – спростити роботу з системою та забезпечити у компактному вигляді звіт з результатів роботи етапів методу. Необхідно надати засоби візуалізації результатів в рамках можливостей: оскільки кількість модальностей може бути довільною, то візуалізація сегментованого простору значень стає неможливою, отже, необхідно досліджувати його перерізи гіперплощинами з постійним значенням деяких модальностей, обмежитись відображенням поділу окремих незалежних величин з залежною, або надавати зведену статистику результатів сегментації та моделювання.

Задача контролера полягає в виконанні проміжної функції, маршрутизації запитів, створених користувачем за допомогою інтерфейсу, приведення їх до необхідного формату та передачі компонентам обробки даних. Так само, в зворотному порядку, контролер приймає результати обробки, перетворює їх у формат, який очікує інтерфейс та передає їх.

Компоненти обробки даних відповідають безпосередньо за алгоритм реалізації методу і аналіз його результатів. За допомогою обраних варіантів реалізації вони виконують побудову, аналіз та використання моделі системи, яка досліджується.

Рівні системи ізольовані та взаємодіють за допомогою спеціально визначених інтерфейсів та протоколів щоб забезпечити їх незалежність. Таким чином, стає можливим використання інших засобів для реалізації рівнів. Також, значно спрощується і перехід до сервіс-орієнтованої архітектури: самі компоненти залишаються практично незмінними, змінюється лише протокол їх взаємодії.

Враховуючи необхідність модифікації реалізації етапів та їх параметрів ізоляція рівнів дозволяє легше впроваджувати, відслідковувати та тестувати зміни в кожному з компонентів, проте, нажаль такі зміни впливають на елементи на всіх рівнях.

Також, можна зазначити, що запропонована архітектура є варіантом розповсюдженого шаблону MVC, різноманітні варіації якого широко застосовується в конструюванні програмного забезпечення для відокремлення інтерфейсу користувача (або в загальному вигляді представлення) від компонентів безпосередньої логіки обробки даних [28].

3.3. Проєктування системи

Визначивши архітектуру, перейдемо до більш детального визначення та проєктування компонентів системи, що розробляється, охарактеризуємо їх призначення і поведінку. Спираючись на отримані результати можна буде обрати засоби розробки для виділених компонентів.

Для реалізації системи застосуємо об’єктно-орієнтований підхід, оскільки його використання дозволяє домогтись виділення окремих інтерфейсів, розширення за рахунок наслідування та перевизначення поведінки в похідних класах або різні реалізації інтерфейсів, а також визначити функціональність системи на всіх рівнях абстракції.

Компоненти обробки даних реалізують етапи запропонованого методу.

Зчитування набору даних

Для зчитування, зберігання та попередньої обробки застосуємо пакети numpy та pandas, які є галузевим стандартом в області аналізу даних завдяки векторизації та реалізації операцій у вигляді попередньо скомпільованого та максимально оптимізованого байт-коду на C.

Набори даних, які підлягають аналізу в більшості випадків в форматі CSV, рідше у вигляді XML або JSON. Зчитування csv- та json-файлів реалізовано у вигляді вбудованих функцій pandas, для зчитування xml-файлів застосуємо парсер lxml, який виконує відображення ієрархічної структури XML в об’єкти:

```
def _read_xml(self, path: str): -> pd.DataFrame
    xml_data = objectify.parse(path)
    root = xml_data.getroot()

    data = []
    columns = []
    for i in range(len(root.getchildren())):
        child = root.getchildren()[i]
        data.append([subchild.text for subchild in
child.getchildren()])
        columns.append(child.tag)

    df = pd.DataFrame(data).T
    df.columns = columns
    return df
```

Підготовка даних

Етап підготовки даних полягає в послідовному застосуванні до набору даних перетворень, які його модифікують.

Послідовність, параметри, передумови та результати роботи кожної процедури наперед невідомі та задаються дослідником, тому доцільно застосувати для класів, що відповідають за етап підготовки даних шаблон «ланцюжок відповідальності», в якому процедури попередньої обробки набору даних реалізовані у вигляді класів-обробників, що дозволяє динамічно налаштувати послідовність таких процедур, задати їм відповідні

параметри, реалізувати логування та аналіз проміжних результатів, тощо, за рахунок стану об’єктів-обробників та перевизначення їх поведінки.

Для створення та налаштування таких обробників варто використати паттерн «фабричний метод», який створює та ініціалізує об’єкти-обробники за їх ідентифікаторами, які надходитимуть із інтерфейсу користувача в результаті його вибору етапів та відповідних параметрів (рис. 3.3) [29].

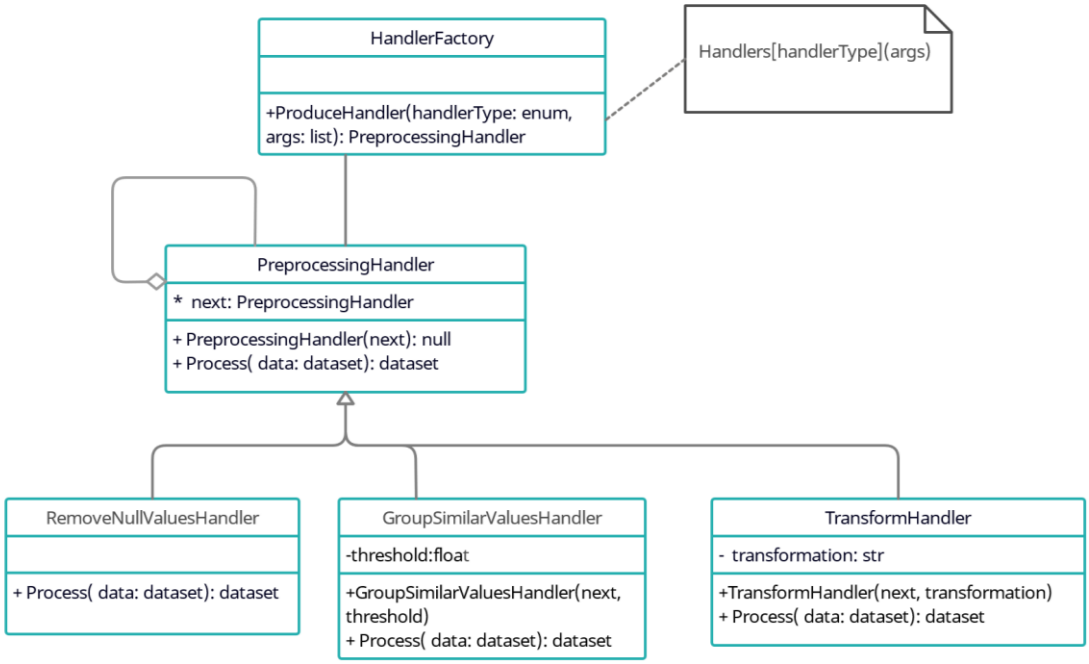


Рис. 3.3. Компоненти попередньої обробки набору даних

Кластеризація

Наступний етап методу – кластеризація над перетвореним набором даних.

Об’єкти що виконують дану функцію приймають результуючий набір даних, та керуючись певним алгоритмом виконують кластеризацію, повертаючи поділ простору значень на інтервали. Таким чином, можна виділити загальний інтерфейс цих об’єктів, реалізуючи який класи надаватимуть різні алгоритми кластеризації (рис 3.4).

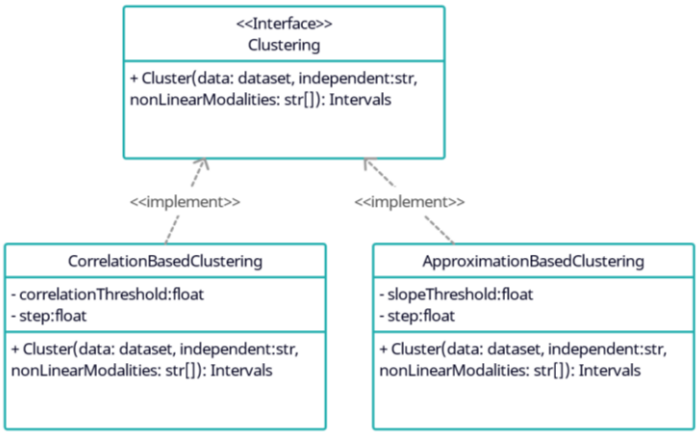


Рис. 3.4. Інтерфейс кластеризації

Наприклад, алгоритм, більш ефективний для розряджених наборів даних, який використовує метод k-means і значення кореляції околах певного розміру (параметр self._range), які перетинаються (параметр self._overlap):

```
def _cluster_modality(self, dataset, dependent, modality):
  partial = dataset[[dependent, modality]].sort_values(by=[modality])
  partial["corr"] = 0.0
  partial = self._append_range_correraltion(partial,dependent,
modality)
  kmeans = KMeans(init="random", n_clusters=self._n, n_init=10)
  kmeans.fit(partial.to_numpy(dtype=np.float32))
  ranges = self._unambiguous_split(kmeans.cluster_centers_,
               dataset[modality].min(), dataset[modality].max())
  return ranges

def _append_range_correraltion(self, dataset, dependent, modality):
  min_value = dataset[modality].min()
  max_value = dataset[modality].max()
  neighbourhood_range = self._range * (max_value - min_value)
  start = min_value
  end = min_value + neighbourhood_range
  while end < max_value:
    subset = dataset[modality >= start-
neighbourhood_range*self._overlap &
                     modality <= end +
neighbourhood_range*self._overlap]
    subset["corr"] = subset[modality].corr(subset[dependent])

    dataset = dataset.join(subset, on="index", how="left")
    start = end
    end += neighbourhood_range

  return dataset
```

Методи моделювання

Для оцінки параметрів моделей застосуємо пакет scikit-learn – бібліотеку, яка містить широкий набір засобів оптимізації, машинного навчання, оцінки моделей та сумісна з інтерфейсами інших пакетів, таких як scipy, numpy і pandas.

Методи моделювання виконують оцінку параметрів моделей на кожному з виділених кластерів за точками набору, що потрапили в нього. Результуючий набір моделей на кластерах використовується для подальшої оцінки значень модальностей та виявлення аномалій.

Для доступу до моделей необхідно організувати їх за допомогою хеш-структури, яка дозволить швидко отримувати необхідну модель за ключем, що будується на основі інтервалів:

```
def fit(self, dataset:, ranges, independent, dependent):
    self._ranges = ranges
    self._index.clear()
    divisions = [list(range(len(l))) for l in ranges.values()]
    for combination in itertools.product(*divisions):
        self._index[combination] = self._fit_model(dataset,
combination,
                                           independent, dependent)

def _fit_model(self, dataset, combination, independent, dependent):
    partial_data = dataset
    i = 0
    for divided_modality in self._ranges.keys():
        partial_data = partial_data.loc[
          (partial_data[divided_modality] >=
           self._ranges[divided_modality][combination[i]][0]) &
          (partial_data[divided_modality] <=
           self._ranges[divided_modality][combination[i]][1])]
        i += 1

    train_x = partial_data[independent]
    train_y = partial_data[dependent]

    if len(train_x.index) > 0:
        regression = linear_model.LinearRegression()
        regression.fit(train_x, train_y)
        return regression
    else:
        return None
```

Так само, під час використання моделі, для точок даних, що надходять, обчислюється ключ і виконується обчислення відповідною моделлю:

```
def predict(self, points: pd.DataFrame):
    points_copy = points.copy()
    points_copy["range_comb"] = points_copy.apply(
                lambda row: self._find_range(row), axis=1)
    points_copy["pred"] = points_copy.apply(
                lambda row: self._predict_row(row), axis=1)
    return points_copy["pred"].to_numpy(dtype=np.float32)
```

Всі етапи методу можна згрупувати в один об’єкт, що зберігає певну конфігурацію етапів. Це дозволяє просто зберегти створений об’єкт за допомогою серіалізації для його повторного використання або порівняння з іншими реалізаціями або методами моделювання. Для його генерації доцільно застосувати шаблон проєктування «будівельник», який керує створенням підоб’єктів, оскільки генерація такого об’єкта за одну операцію складна. Також слід врахувати, що керування створенням цього об’єкту відбуватиметься через графічний інтерфейс, що також потребує забезпечення спеціального механізму конструювання [29].

Контролер – це підсистема, яка отримує дані з інтерфейсу користувача та на їх основі формує та відправляє запити до компонентів обробки, які реалізують логіку методу і навпаки, отримавши результати виконання етапів або їх аналізу генерує результати, які відображатимуться на користувацькому інтерфейсі.

Запити до контролера генеруються подіями, що відбуваються на інтерфейсі, такими як, наприклад, натискання на кнопки або завантаження екрану. Контролер надає рівень абстракції між інтерфейсом та логікою методу і слугує адаптером між ними, перетворюючи дані у формат, прийнятний для інших компонентів.

При цьому, інтерфейси, які взаємодіють між собою є численними, тому для їх обробки доцільно використати окремі адаптери, які займаються певним обмеженим набором логічно пов’язаних зв’язків і

агреговані в спільному пакеті, який делегує запити спеціалізованим підоб'єктам, щоб розподілити відповідальність та зменшити об'єм як коду, так і функціональності, покладений на кожну окрему сутність.

Проєктування інтерфейсу будується на основі функціональних вимог (рис. 3.1) та сценаріїв використання, що відповідають динамічній моделі системи (рис. 3.5).

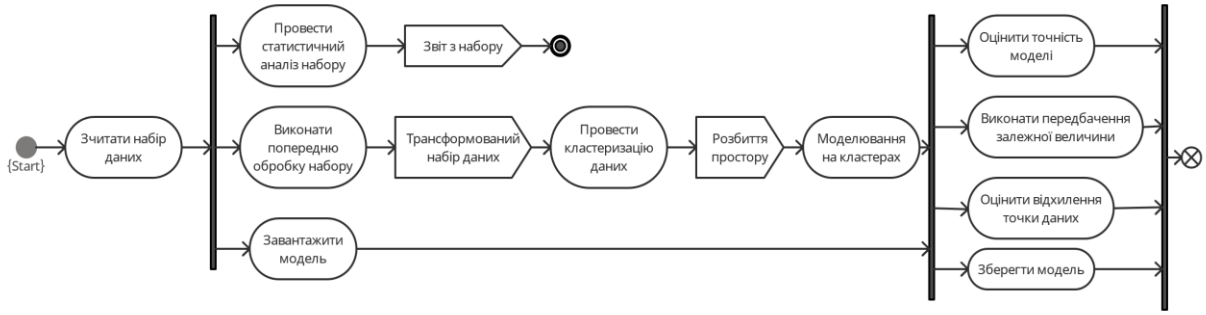


Рис. 3.5. Діаграма діяльності системи

Інтерфейс користувача надає зрозумілі засоби, які реалізують дані функції і складається з екранів, які дозволяють користувачу більш зрозуміло взаємодіяти з системою:

- Головний екран.

  Початковий екран програми, який дозволяє завантажити набір даних, який буде досліджуватись та надає стислу його характеристику. Також головний екран дозволяє згенерувати результати кореляційного аналізу набору в форматі pdf і обрати з набору даних модальності, які використовуються при побудові моделі та залежну величину.

- Екрани етапів відображають результати виконання етапів методу для аналізу та верифікації.
  - Екран попередньої обробки дозволяє обрати перетворення, які будуть виконані над набором даних, дозволяє налаштувати їх параметри (за наявності) завдяки допоміжним конфігураційним

екранам та відображає стислу інформацію про набір даних після застосування процедур обробки.

- Екран кластеризації дозволяє обрати метод кластеризації та його параметри і відображає результати поділу простру мультимодальних даних за їх статистичними характеристиками, розподіл точок в них.
- Екран моделювання відображає загальну інформацію про створені на інтервалах простору моделі (їх кількість, кількість параметрів), а також відображає оцінку точності моделі за тестовим набором даних. Параметрами оцінки точності є середнє квадратичне відхилення (як загально прийнята метрика точності подібних моделей) та середнє абсолютне відхилення (окрім метрики точності моделі, цей параметр визначає середню нев'язку, тобто вказує, яке відхилення є допустимим в рамках похибки моделювання і тому слугує показником для визначення аномалій: кортежі, що значно перевищують це відхилення можна вважати аномальними). Окрім цього, модель можна зберегти для подальшого дослідження і порівняння, а також виконати обчислення залежної величини чи перевірити відхилення повного кортежу даних.

- Екрани конфігурації.

  Вони є службовими елементами і дозволяють обрати параметри для реалізацій етапів. Кожна процедура, що застосовується до набору даних має свій ряд параметрів, які можна налаштувати для отримання більш точного результату через відповідний компактний екран. Параметри, що задаються на цих екранах визначаються самою процедурою, в тому числі для перетворень, що не мають параметрів (наприклад, видалення тотожних кортежів) екран конфігурації не передбачається.

Для реалізації інтерфейсу використовується wxPython – бібліотека для створення кросплатформних графічних інтерфейсів користувача, яка дозволяє швидко створювати програми з надійним, функціональним віконним інтерфейсом, який відповідає вигляду елементів цільової платформи. Дана бібліотека є надбудовою над wxWidgets, яка дозволяє використовувати програмний інтерфейс Python.

3.4. Висновки до розділу III

Даний розділ присвячено проєктуванню програмного забезпечення, що реалізує запропонований метод поєднання мультимодальних даних.

Було сформульовано та класифіковано вимоги до розроблюваного програмного забезпечення. На основі запропонованих вимог було визначено архітектурні шаблони, які можна використати для реалізації та тестування системи, серед яких виділено багаторівневу та міркосервісну архітектуру. В рамках даної роботи система буде створюватись з використанням багаторівневої архітектури, через простоту реалізації та тестування монолітних системи у порівнянні з сервіс-орієнтованими, проте в процесі розвитку та роботи з великим обсягом даних можливий перехід на сервіс-орієнтовану архітектуру з метою використання декількох інстансів компонентів системи у вигляді сервісів, доступних у мережі.

Було описано призначення та зв’язки компонентів системи, їх функціональність та механізми взаємодії. Зважаючи на прийняті архітектурні рішення та специфіку системи було обрано засоби розроблення програмного забезпечення, які широко застосовуються для ефективного розв’язання подібних задач: мова програмування Python, яка найчастіше застосовується для інтелектуального аналізу даних та машинного навчання, пакети numpy, pandas, scikit-learn для обробки даних, моделювання та верифікації, wxPython та matplotlib для створення графічних інтерфейсів та візуалізації за допомогою діаграм і графіків.

4. АНАЛІЗ ОТРИМАНИХ РЕЗУЛЬТАТІВ

4.1. Застосування запропонованого методу

Розглянемо приклад застосування запропонованого методу для аналізу набору даних.

4.1.1. *Характеристика та аналіз набору даних*

Набір даних, що розглядається отримано з дослідження енергоефективності житлових будівель, тобто задачею моделювання є прогнозування кількості енергії, яка необхідна для підтримання допустимої температури.

Будівлі для яких проводились вимірювання знаходяться в Афінах, Греція, та мають однаковий об'єм в 771,75 $м^3$ та побудовані з використанням однакових матеріалів, але мають різні розміри, орієнтацію та кількість вікон та їх положення.

Набір даних містить вісім незалежних змінних:

- Відносна компактність – важлива характеристика енергоефективності, яка обчислюється як спеціальна міра відношення об'єму приміщення до суми площі поверхонь, які контактують із зовнішнім середовищем $R_c = 6 * V^{0.66} * (\sum S_{ext})^{-1}$.
- Площа поверхні (м2).
- Площа бічних стін (м2).
- Площа даху (м2).
- Висота будівлі (м).
- Орієнтація будівлі: 2 – північ, 3 – схід, 4 – південь, 5 – захід.
- Площа вікон відносно площі стін (%).
- Розміщення вікон: 1 – рівномірно з усіх сторін будівлі, 2 – 55% з виходом на північ інші – по 15% в інші сторони, аналогічно 3 – 55% на схід, 4 – 55% на південь, 5 – 55% на захід.

Також, набір містить дві залежні величини, обчислені для відповідних будівель – теплове та охолоджувальне навантаження, тобто кількість енергії, яку необхідно додати або відвести з приміщення, щоб підтримувати температуру в межах допустимих значень під час холодного або теплого сезону відповідно (кВт*год./м^2).

Набір даних міститься у csv-файлі, в якому незалежні змінні позначені X1-X8 відповідно, залежні – Y1, Y2, та містить 768 точок даних, не містить відсутніх даних або тотожних кортежів [30].

Розглянемо статистичні характеристики значень в наборі даних, наведені в табл. 4.1 (X6 та X8 є дискретними значеннями, які використовуються для кодування, тому для них недоцільно обчислення тих самих статистичних величин, як для інших модальностей).

Таблиця 4.1

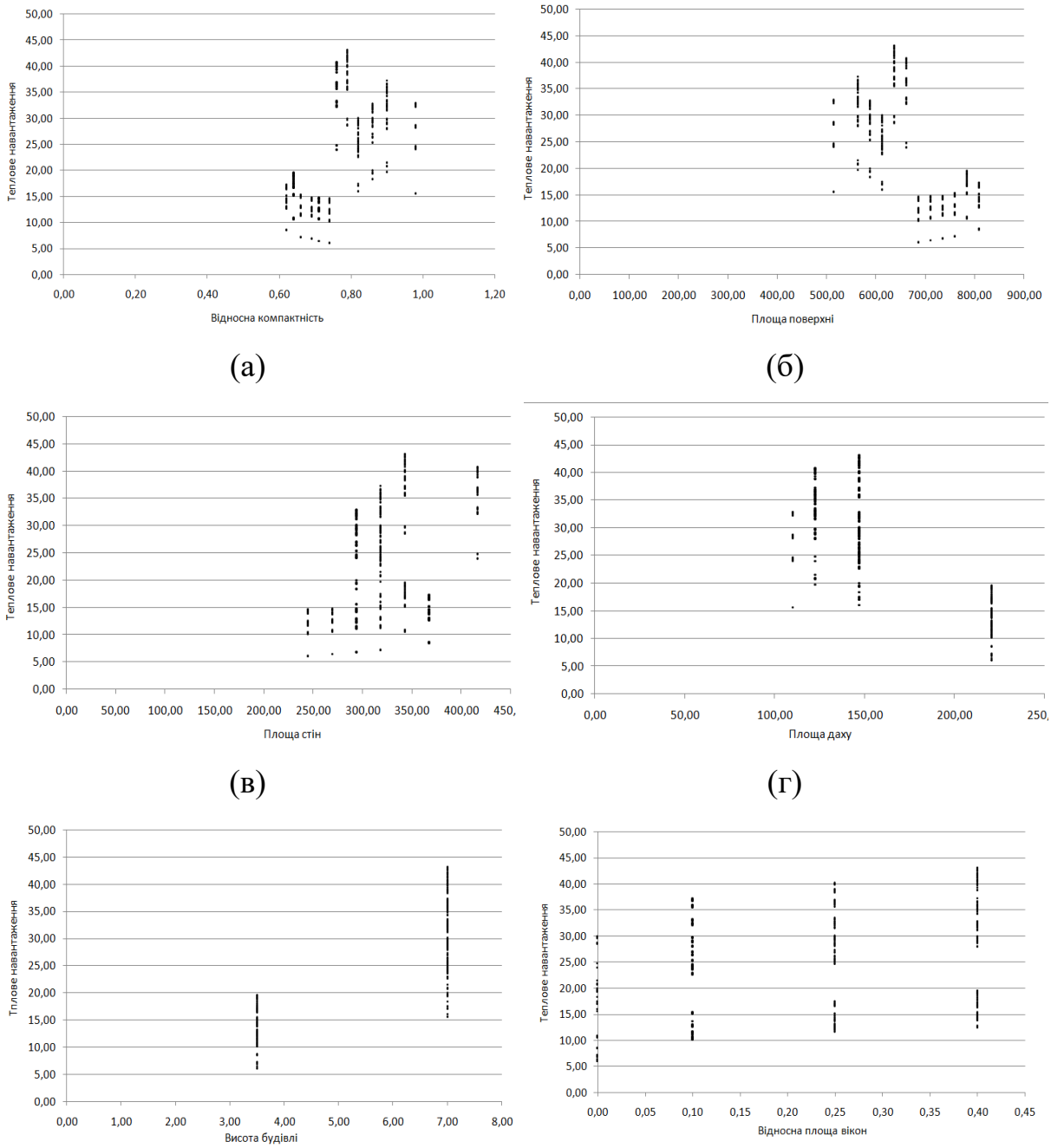
Статистичні характеристики модальностей

	X1	X2	X3	X4	X5	X7	Y1	Y2
Діапазон значень	0.62-0.98	514.5-808.5	245.0-416.5	110.25-220.5	3.5-7.0	0.0-0.4	6.01-43.1	10.9-48.03
Кількість унікальних	12	12	7	4	2	4	586	636
Середнє значення	0.764	671.7	318.5	176.6	5.25	0.234	22.31	24.588
Дисперсія	0.011	7759.17	1903.27	2039.97	3.066	0.018	101.812	90.503

Отримані характеристики показують, що відносно невелика кількість унікальних значень незалежних величин призводить до значної варіації в залежних, що означає наявність впливу на залежні величини, які мають великі (порівняно з середнім) значення дисперсії. Також, можна припустити, що ця варіація викликана величинами з більшими відхилень.

Розглянемо також кореляцію між модальностями та залежність між незалежними і залежними величинами (рис 4.1, табл. 4.2, 4.3). Це

дозволить виявити певні закономірності в даних, які можна використати для більш точного моделювання системи.



(а) (б)

(в) (г)

(д) (е)

Рис. 4.1. Залежності між незалежними величинами і тепловим навантаженням (Y1): (а) відносна компактність (X1); (б) площа поверхні (X2); (в) площа бічних стін (X3); (г) площа даху(X4); (д) висота будівлі (X5); (е) відносна площа вікон (X7)

Таблиця 4.2

Кореляція Пірсона між модальностями

	X1	X2	X3	X4	X5	X6	X7	X8	Y1
X1	1.00	–0.99	–0.20	–0.87	0.83	0.00	0.00	0.00	0.62
X2	–0.99	1.00	0.20	0.88	–0.86	0.00	0.00	0.00	–0.66
X3	–0.20	0.20	1.00	–0.29	0.28	0.00	0.00	0.00	0.46
X4	–0.87	0.88	–0.29	1.00	–0.97	0.00	0.00	0.00	–0.86
X5	0.83	–0.86	0.28	–0.97	1.00	0.00	0.00	0.00	0.89
X6	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00
X7	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.21	0.27
X8	0.00	0.00	0.00	0.00	0.00	0.00	0.21	1.00	0.09
Y1	0.62	–0.66	0.46	–0.86	0.89	0.00	0.27	0.09	1.00

Таблиця 4.3

Рангова кореляція Спірмена між модальностями

	X1	X2	X3	X4	X5	X6	X7	X8	Y1
X1	1.00	–1.00	–0.26	–0.87	0.87	0.00	0.00	0.00	0.62
X2	–1.00	1.00	0.26	0.87	–0.87	0.00	0.00	0.00	–0.62
X3	–0.26	0.26	1.00	–0.19	0.22	0.00	0.00	0.00	0.47
X4	–0.87	0.87	–0.19	1.00	–0.94	0.00	0.00	0.00	–0.80
X5	0.87	–0.87	0.22	–0.94	1.00	0.00	0.00	0.00	0.86
X6	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00
X7	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.19	0.32
X8	0.00	0.00	0.00	0.00	0.00	0.00	0.19	1.00	0.07
Y1	0.62	–0.62	0.47	–0.80	0.86	0.00	0.32	0.07	1.00

Отримані результати дозволяють зробити ряд висновків про приховані закономірності в даних:

- Значення залежної величини зосереджені вздовж певних унікальних значень незалежних величин, наявних в наборі, але мають помітну варіацію, що означає наявність суттєвого впливу інших величин на значення теплового навантаження.
- Залежності відносної компактності та площі поверхонь, що контактують з навколишнім середовищем мають нелінійну залежність з залежною величиною і точки даних цих модальностей розміщені групами з високою густиною.
- Залежна величина має сильну кореляцію Пірсона з X5 та X4, меншу з X1, X2. Це означає відносно високу міру лінійної залежності між ними. Схожа ситуація складається із кореляцією Спірмена, яка вказує на монотонну залежність між величинами, хоча зважаючи на розподіл точок даних коефіцієнти кореляції не є чіткими показниками, оскільки є чутливими до викидів із загального розподілу, якими в даному випадку є групи точок даних.

4.1.2. *Застосування єдиної лінійної моделі*

Лінійна регресія є поширеним, дослідженим і ефективно реалізованим методом моделювання систем. Застосуємо множинну регресію для набору даних, що розглядається:

$$Y_1 = \sum_{i=1}^{8} a_i X_i + a_0$$

Оцінка параметрів моделі за допомогою методу найменших квадратів дає наступні оцінки параметрів (табл. 4.4).

Таблиця 4.4

Оцінка параметрів регресійної моделі

a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8
80.57	–62.213	–5744.7	5744.7	11489.22	4.227	–0.054	19.65	0.21

Для обчислення параметрів було використано 90% наявних в наборі даних і 10% було випадково відібрано для верифікації моделі. Оцінка моделі на тестових даних дає наступні результати:

- Середня квадратична помилка: 9.89.
- Середня абсолютна помилка: 2.33.
- Коефіцієнт детермінації: 0.89.

Отримані результати показують, що створена модель є хорошим наближенням системи. Високий коефіцієнт детермінації означає що варіація залежної величини майже повністю пояснюється набором незалежних, а середня абсолютна помилка моделі відносно мала в порівнянні зі стандартним відхиленням залежної величини.

Проаналізуємо отримане наближення. Отримані коефіцієнти свідчать, що змінні X6, X8 мають дуже незначний вплив на значення залежної величини, що не дивно, враховуючи що це дискретні значення, які кодують певне розміщення і безпосередні значення кодів обрано довільно з розсуду дослідника, який отримав набір даних (тобто, їх значення можна довільно обрати іншим чином, що робить неможливим їх змістовну інтерпретацію як незалежних змінних моделі).

Враховуючи це, побудуємо модель без їх урахування (табл. 4.5).

Таблиця 4.5

Оцінка параметрів регресійної моделі без дискретних значень

a_0	a_1	a_2	a_3	a_4	a_5	a_7
79.28	–61.446	–6022.94	6022.92	12045.71	4.199	20.15

Метрики точності за однакових умов обчислення параметрів та верифікації моделі:

- Середня квадратична помилка: 9.78.
- Середня абсолютна помилка: 2.25.
- Коефіцієнт детермінації: 0.89

Значення помилок моделі дещо зменшились, але в даному представленні модель оперує лише змістовними параметрами, які можна інтерпретувати, як реальні фізичні показники, що впливають на залежну величину і визначаються об'єктивними вимірами.

4.1.3. *Застосування запропонованого методу*

Розглянутий набір даних є розрідженим, тому для нього можна використати кластеризацію на основі коефіцієнта кореляції.

Так само як і в експериментах із застосуванням одного лінійного наближення, не виконуватимемо ніякої додаткової обробки набору даних.

Враховуючи будову набору даних, доцільно застосувати кластаризацію до X1 та X2, оскільки вони достатньо розподілені в просторі та мають нелінійну залежність з Y1. В наступному експерименті кластеризація застосовується також до X3. Для інших модальностей поділ простору значень недоцільний, оскільки вони мають дуже малу кількість унікальних значень і поділ простору призведе до перенавчання моделі на цих ізольованих інтервалах і знизить точність і надійність моделі.

Результати кластеризації наведено на рис. 4.2, 4.3.

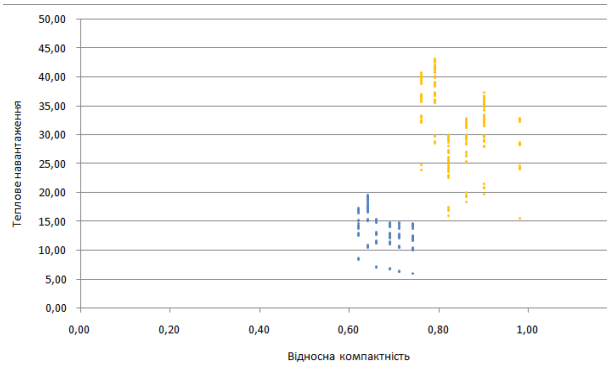


Рис. 4.2. Кластеризація X1

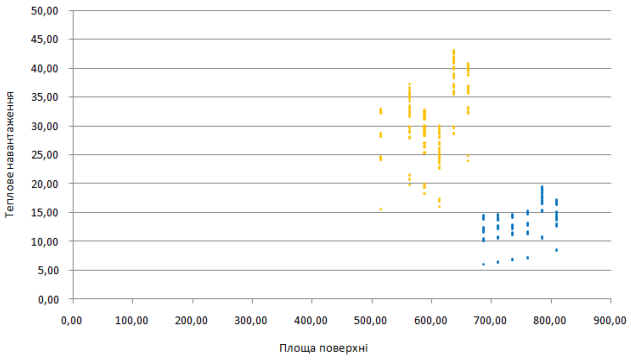


Рис. 4.3. Кластеризація X2

Таким чином маємо два поділи простору гіперплощинами X1 = 0.75 та X2 = 662. Без подальшого поділу простору іншими модальностями необхідно обчислити чотири моделі. Результати обчислення параметрів наведено в табл. 4.6.

Таблиця 4.6

Параметри моделей після поділу простору (X1, X2)

№	Поділ	Параметри моделі								
1	X1 < 0.75, X2 < 622	–								
2	X1 ≥ 0.75, X2 < 622	a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8
		–23.04	14.18	0.021	0.023	$-6\ 10^{-8}$	$4.47 \cdot 10^{-8}$	$-8 \cdot 10^{-8}$	14.26	0.114
3	X1 ≥ 0.75, X2 ≥ 622	–								
4	X1 < 0.75, X2 ≥ 622	a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8
		–1716.2	936.1	3875	–3874	–7746.9	8.45	–0,057	25.58	0.29

Можна помітити, що коефіцієнти моделей та вплив змінних сильно різняться між моделями, що дає підставу інтерпретувати отриманий поділ як деякі стани системи, в яких поведінка та значення величин різняться.

Також можна зауважити, що тенденції, наявні в єдиній лінійній моделі частково проявляються лише в деяких моделях на кластерах, наприклад протилежний вплив X2 та X3 у випадку моделі на підпросторі 4.

У підпростори 1 та 3 не потрапила жодна точка через фізичний зв’язок цих величин (вони майже обернено пропорційні з визначення відносної компактності), тому там неможливо побудувати будь-яку апроксимацію.

Метрики точності побудованої моделі значно кращі, ніж за використання одного наближення, що дозволяє стверджувати, що вона більш точно відповідає наявним даним:

- Середня квадратична помилка: 4.78.
- Середня абсолютна помилка: 1.73.
- Коефіцієнт детермінації: 0.95.

Розглянемо також розглянемо результати моделювання з кластеризацією модальності X3 (рис. 4.4, табл. 4.7) та виключенням дискретних модальностей X6 та X8 (табл. 4.8).

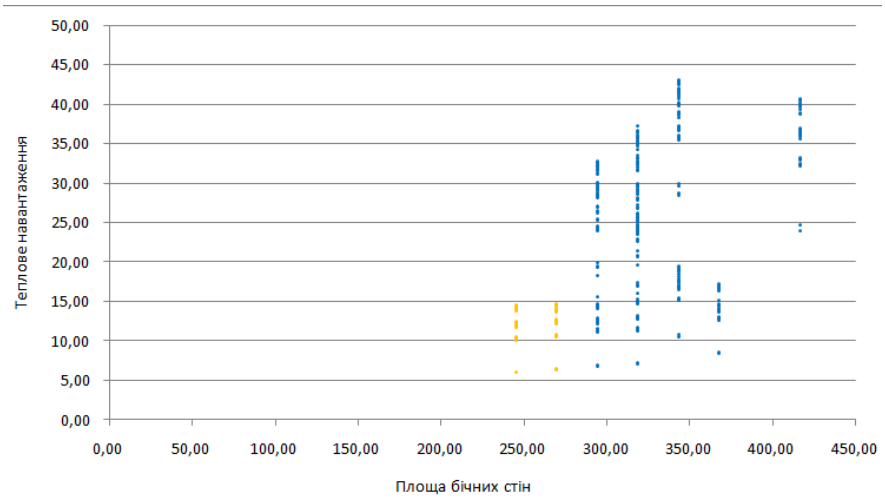


Рис. 4.4. Кластеризація X3

Таблиця 4.7

Параметри моделей після поділу простору (X1, X2, X3)

№	Поділ	Параметри моделі								
1	X1 < 0.75, X2 < 622, X3 < 270	–								
2	X1 < 0.75, X2 < 622, X3 ≥ 270	–								
3	X1 < 0.75, X2 ≥ 622, X3 < 270	a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8
		5.23	$-3.9\ 10^{-6}$	$3.13\ 10^{-3}$	$-3.1\ 10^{-3}$	$-4.8\ 10^{-7}$	$-1.9\ 10^{-6}$	$1.39\ 10^{-2}$	14.32	0.118

Продовження табл. 4.7

4	X1 < 0.75, X2 ≥ 622, X3 ≥ 270	a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8
		78.63	–71.63	$-2.7\ 10^{-2}$	$-1.1\ 10^{-3}$	$-1.55\ 10^{-6}$	$8\ 10^{-7}$	$5.11\ 10^{-3}$	14.23	0.112
5	X1 ≥ 0.75, X2 < 622, X3 < 270	–								
6	X1 ≥ 0.75, X2 < 622, X3 ≥ 270	a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8
		–1716.24	936.11	3875	–3873.56	–7746.9	8.46	–0.057	25.58	0.29
7	X1 ≥ 0.75, X2 ≥ 622, X3 < 270	–								
8	X1 ≥ 0.75, X2 ≥ 622, X3 ≥ 270	–								

Метрики точності даної моделі:

- Середня квадратична помилка: 5.35.
- Середня абсолютна помилка: 1.72.
- Коефіцієнт детермінації: 0.95.

Можемо зробити висновок, що використання поділу за X3 не є доцільним, оскільки фактично додає лише одну додаткову модель, при чому оцінка точності стає гіршою, а вплив змінних на залежну величину сильно коливається між станами.

Наостанок розглянемо модель з поділом X1, X2 та без дискретних значень кодів (табл. 4.8).

Метрики точності даної моделі:

- Середня квадратична помилка: 4.77.
- Середня абсолютна помилка: 1.60.
- Коефіцієнт детермінації: 0.95.

Для даної моделі маємо найкращі оцінки точності з усіх побудованих.

Таблиця 4.8

Параметри моделей після поділу простору (X1, X2) без дискретних кодів

№	Поділ	Параметри моделі						
1	X1 < 0.75, X2 < 622	–						
2	X1 < 0.75, X2 ≥ 622	a_0	a_1	a_2	a_3	a_4	a_5	a_7
		–22.75	14.18	0.02	0.02	0	0	14.55
3	X1 ≥ 0.75, X2 ≥ 622	–						
4	X1 < 0.75, X2 ≥ 622	a_0	a_1	a_2	a_3	a_4	a_5	a_7
		–1652.02	933.25	3765.53	–3764.01	–7527.98	0	26.31

4.1.4. *Підсумки проведених експериментів*

В результаті застосування запропонованого методу було отримано різні моделі для досліджуваного набору даних, які мають кращі метрики точності ніж відповідні їм за використаними змінними моделі, що використовують одне лінійне наближення для всього набору даних.

Отримані моделі можна інтерпретувати як різні стани системи, які апроксимуються різними наближеннями і в яких спостерігається сильніший вплив для різних змінних.

Доцільним є поділ не всіх модальностей, особливо в сильно розрідженому наборі даних який містить невелику кількість унікальних значень незалежних величин. Так, у випадку з поділом за X1, X2, X3 маємо поділ простору на вісім частин, з яких дані є (і відповідно можна провести моделювання) лише у трьох.

Отримана модель може бути використана для точної оцінки енергоефективності будівель за заданими розмірами і, відповідно, її оптимізації при проєктуванні. Виявлення аномалій в кортежах даних в даному контексті не є дуже корисним, хоча дозволяє зробити висновок про

більші ніж передбачається витрати тепла, що може свідчити про проблеми з утепленням будівлі.

4.2. Особливості запропонованого методу

Важливою характеристикою методу є його головна ідея – пошук інтервалів простору, визначеного вхідним набором мультимодальних даних, на яких певні наближення, наприклад лінійні, будуть більш точними за допомогою емпіричних статистичних характеристик даних, таких як коваріація та кореляція.

Логічним узагальненням методу є також припущення про застосування абстрактних (не тільки лінійних) наближень і пошуку інтервалів, на яких вони достатньо точно моделюють систему. В цьому плані лінійні наближення є кращими, оскільки існують статистичні ознаки, за якими можна легко встановити лінійну залежність між величинами, а методи обчислення параметрів моделей за наявними даними (МНК та стохастичний градієнтний спуск при великій кількості даних та модальностей) є ефективними як з теоретичної так і з практичної точки зору [31], тому в даному дослідженні розглядались саме вони.

Описаний підхід дозволяє застосовувати його до систем зі складною нелінійною поведінкою, або до тих, що поводяться майже лінійно, але по-різному в залежності від певних внутрішніх характеристик (стану) з більшою точністю, ніж такі методи, як регресійні моделі або методи факторизації.

Втім, це справедливо, для систем, інваріантних в часі, тобто таких, які не змінюють (або дуже мало змінюють) поведінку з часом, наприклад, такі як розглянута вище енергоефективність будівель, оскільки з плином часу системи довільно змінюють поведінку, породжуючи велику кількість інтервалів, які можна лінійно апроксимувати. Запропонований метод є менш точним в такому разі і надати змістовну характеристику отриманим результатам досить складно. Його ефективність можна покращити,

використавши час або деякі його складові (наприклад такі як година дня або місяць року) як додаткову модальність, що допоможе в аналізі систем, циклічних відносно обраних часових характеристик. Циклічність систем можна виявити на етапі аналізу набору даних за допомогою таких засобів, як автокореляція з різними затримками або частотний аналіз.

Також, метод використовується для створення статичної моделі системи, тобто після створення модель не модифікується. В той час для цифрових двійників деяких систем є корисною та необхідною їх взаємодія із системою, що моделюється.

Як і будь-який метод інтелектуального аналізу даних, запропонований метод містить велику кількість параметрів – характеристик, які впливають на його ефективність, але не можуть бути оцінені чи наближені апріорно, а тільки виявлені дослідником експериментальним шляхом, тому для досягнення бажаної точності необхідно проведення певної кількості спроб і помилок, що ускладнюється тим, що метод є більш обчислювально складним, ніж застосування однієї моделі для всього набору даних. Втім, абстрактне формулювання етапів методу дозволяє досліднику підбирати конкретні реалізації з урахуванням особливостей набору даних та задачі моделювання, що також дає можливість розглядати кожен етап як окрему задачу та знаходити для неї задовільні рішення.

Так, в загальному випадку якщо поділ застосовано для N модальностей на k_1, k_2, ..., k_N інтервалів, то необхідно обчислення $\prod_{i=1}^{N} k_i$ моделей для кожного з інтервалів. Окрім цього, при використанні моделі для прогнозування також спочатку необхідно визначити до якого інтервалу належить точка даних і обрати відповідну модель, що хоч і не сильно при використанні ефективних алгоритмів пошуку, але все одно сповільнює процес обчислень.

Як було показано вище в проведених експериментах, якщо набір даних розріджений, то велика кількість інтервалів можуть бути пустими.

Ще гірша ситуація складається у випадку, коли в інтервали потрапляє невелика кількість даних, що призводить до перенавчання моделі і гіршої точності в даному інтервалі. В такій ситуації рекомендується поєднати невеликі інтервали з сусідніми щоб зменшити кількість обчислень та запобігти перенавчанню моделей.

Сконструйована за допомогою даного методу модель потребує додаткового аналізу та інтерпретації, а саме потрібно надати певного фізичного сенсу виділеним кластерам та відповідним апроксимаціям. Вони можуть бути корисними в виявленні прихованих закономірностей між модальностями, особливо монотонних нелінійних залежностей, які апроксимуються лінійними фрагментами.

4.3. Напрямки вдосконалення та подальша робота

Напрямки вдосконалення та модифікації безпосередньо самого методу полягають в його особливостях, розглянутих вище.

В першу чергу, це формулювання рекомендацій та/або способів визначення параметрів методу на основі наявного набору даних. Маючи їх оцінки можна значно зменшити кількість ресурсів, часових та обчислювальних, необхідних для побудови моделі із задовільною точністю. Також, вони несуть в собі додаткову інформацію про поведінку системи, яку можна інтерпретувати та використати при її дослідженні, в тому числі в спробах визначити приховані зв’язки між модальностями більш точно, ніж за допомогою апроксимацій, отриманих в результаті роботи методу.

Також важливим доповненням було б створення можливості модифікації результатів за рахунок обробки нових даних, що надходять зі спостережень, створюючи цикл, в якому реальна система та її модель співіснують паралельно і модель оновлюється в залежності від змін в реальному світі.

З точки зору програмної реалізації покращенням може бути використання розподіленого виконання етапів методу, що дозволить обробляти великі набори даних за прийнятний час. Допомогти в цьому може в тому числі і перехід на мікросервісну архітектуру, і розгортання компонентів програмного забезпечення в декількох екземплярах на одній або декількох машинах, пов’язаних мережею. Впровадження такого підходу вимагає додатково використання планувальника – підсистеми, яка розподілятиме задачі між доступними сервісами та виконуватиме агрегацію результатів.

Також можна створити кращі засоби візуалізації та аналізу результатів, які допоможуть в дослідженні систем та оцінці отриманої моделі та проміжних результатів етапів методу.

4.4. Висновки до розділу IV

В даному розділі було проведено дослідження запропонованого методу поєднання мультимодальних даних, виявлено деякі його особливості та характеристики, запропоновано напрямки подальшої роботи та вдосконалення методу для їх покращення.

За результатами проведених досліджень метод показує кращу точність для інваріантних в часі систем, ніж використання одного наближення за рахунок виділення інтервалів простору значень, в яких це наближення буде достатньо точним за статистичними характеристиками даного інтервалу. В дослідженні розглядались лінійні наближення, тому такими характеристиками є коефіцієнти кореляції.

Розглянутий метод є точним на розріджених наборах даних та на наборах, в яких спостерігається кусково-монотонна залежність між досліджуваними величинами, оскільки тоді ідея виділення інтервалів, на яких певна апроксимація буде більш точною дає непогані результати.

Таким чином, серед переваг створеного методу можна виділити точність моделювання, яка перевищує точність інших методів, які часто

застосовуються для розв’язання подібних задач і використовують єдину модель, серед недоліків – обчислювальну та просторову складність та необхідність визначення найбільш підходящих реалізацій етапів та їх параметрів експериментально.

Подальша робота буде направлена на виправлення недоліків методу та програмної реалізації, а також на створення нових, більш ефективних засобів обробки та візуалізації проміжних результатів. Також, необхідна модифікація методу, яка дозволить використання дискретних значень в моделюванні, оскільки вони часто є визначними в виявленні станів системи.

ВИСНОВКИ

Дана робота присвячена створенню методу поєднання мультимодальних даних для створення цифрових двійників за набором даних.

В ході дослідження було розглянуто існуючі методи поєднання мультимодальних даних та виявлено їх спільний недолік: застосування однієї лінійної апроксимації для всього набору даних, що є неефективним у випадку, коли система, що моделюється, підкоряється нелінійним законам або змінює поведінку в залежності від стану, який визначається деяким набором параметрів.

Для покращення точності моделювання було запропоновано метод поєднання даних, який використовує кластеризацію за статистичними характеристиками даних та використання декількох моделей на виділених кластерах. Таки підхід дозволяє виділити стани системи в наборі даних та застосувати різні апроксимації для їх опису. Навіть якщо поведінка системи не змінюється в залежності від стану запропонований метод дозволяє отримати виграш в точності за рахунок кращого чисельного наближення на інтервалах простору даних.

Запропонований метод є проміжним між керованими даними та керованими моделями методами, оскільки передбачає використання певної наперед визначеної моделі для апроксимації, але використовує закономірності в даних, щоб забезпечити більшу точність даної моделі.

Можливість обрати реалізацію етапів методу для досягнення більшої точності дозволяє адаптувати його до широкого спектру задач, та дослідження свідчать, що він найбільш ефективний для розріджених наборів даних інваріантних в часі систем.

Було створено програмне забезпечення, яке реалізує метод. З його використанням метод було порівняно з методом, який часто використовується для моделювання – множинною лінійною регресією.

Порівняння на наборі даних демонструє, що запропонований метод дозволяє отримати більшу точність моделі, але є більш обчислювально складним.

Подальший розвиток методу буде направлений на покращення його недоліків, а саме: статичність (модель неможливо оновлювати паралельно з реальною системою), та використання неперервних даних. Хоча дискретні значення можна використати для моделювання, застосувавши змістовне кодування або перетворення на початковому етапі, безпосередньо сам метод не використовує дискретні значення при моделюванні, хоча вони можуть містити корисну інформацію про поведінку системи.

СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. Lahat D., Adali T., Jutten C. Multimodal data fusion: an overview of methods, challenges, and prospects //Proceedings of the IEEE. – 2015. – Т. 103. – №. 9. – С. 1449-1477.
2. Digital twin: towards a meaningful framework [Електронний ресурс]. – Режим доступу: https://www.arup.com/perspectives/publications/research/section/digital-twin-towards-a-meaningful-framework – Дата доступу: лютий 2021.
3. Gao J. et al. A survey on deep learning for multimodal data fusion //Neural Computation. – 2020. – Т. 32. – №. 5. – С. 829-864.
4. Cai Y. et al. Sensor data and information fusion to construct digital-twins virtual machine tools for cyber-physical manufacturing //Procedia manufacturing. – 2017. – Т. 10. – С. 1031-1042.
5. Qi Q., Tao F. Digital twin and big data towards smart manufacturing and industry 4.0: 360 degree comparison //Ieee Access. – 2018. – Т. 6. – С. 3585-3593.
6. Cimino C., Negri E., Fumagalli L. Review of digital twin applications in manufacturing //Computers in Industry. – 2019. – Т. 113. – С. 103130.
7. Enders M. R., Hoßbach N. Dimensions of digital twin applications-a literature review. – 2019.
8. Liu M. et al. Review of digital twin about concepts, technologies, and industrial applications //Journal of Manufacturing Systems. – 2020.
9. Barricelli B. R., Casiraghi E., Fogli D. A survey on digital twin: Definitions, characteristics, applications, and design implications //IEEE Access. – 2019. – Т. 7. – С. 167653-167671.
10. Rasheed A., San O., Kvamsdal T. Digital twin: Values, challenges and enablers //arXiv preprint arXiv:1910.01719. – 2019.

11. A Step-by-Step Explanation of Principal Component Analysis (PCA). [Електронний ресурс]. – Режим доступу: https://builtin.com/data-science/step-step-explanation-principal-component-analysis – Дата доступу: січень 2021.
12. Raol J. R. Data fusion mathematics: theory and practice. – CRC Press, 2015.
13. Bro R. PARAFAC. Tutorial and applications //Chemometrics and intelligent laboratory systems. – 1997. – Т. 38. – №. 2. – С. 149-171.
14. Castanedo F. A review of data fusion techniques //The scientific world journal. – 2013. – Т. 2013.
15. The Probabilistic Digital Twin Concept [Електронний ресурс]. – Режим доступу: https://ai-and-safety.dnvgl.com/probabilistic-twin/probabilistictwin.html – Дата доступу: грудень 2020.
16. Meng T. et al. A survey on machine learning for data fusion //Information Fusion. – 2020. – Т. 57. – С. 115-129.
17. Lek S., Park Y. S. Artificial neural networks //Encyclopedia of Ecology, Five-Volume Set. – Elsevier Inc., 2008. – С. 237-245.
18. Multimodal Deep Learning. Fusion of multiple modalities using Deep Learning [Електронний ресурс]. – Режим доступу: https://towardsdatascience.com/multimodal-deep-learning-ce7d1d994f4 – Дата доступу: вересень 2020.
19. Chu X. et al. Data cleaning: Overview and emerging challenges //Proceedings of the 2016 international conference on management of data. – 2016. – С. 2201-2206.
20. Norris S. Systematically working with multimodal data: Research methods in multimodal discourse analysis. – John Wiley & Sons, 2019.
21. Burton A., Altman D. G. Missing covariate data within cancer prognostic studies: a review of current reporting and proposed guidelines //British journal of cancer. – 2004. – Т. 91. – №. 1. – С. 4-8.

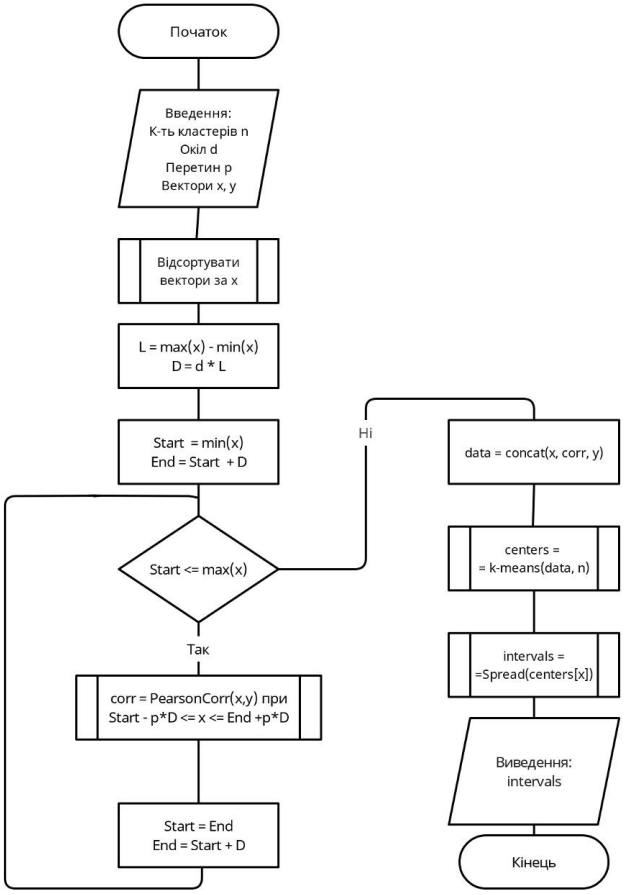
22. Horton N. J., Kleinman K. P. Much ado about nothing: A comparison of missing data methods and software to fit incomplete data regression models //The American Statistician. – 2007. – Т. 61. – №. 1. – С. 79-90.
23. Jan J. Digital signal filtering, analysis and restoration. – IET, 2000. – №. 44.
24. Namey E. et al. Data reduction techniques for large qualitative data sets //Handbook for team-based qualitative research. – 2008. – Т. 2. – №. 1. – С. 137-161.
25. Bevington P. R., Robinson D. K. Data reduction and error analysis //McGraw-Hill, New York. – 2003.
26. How to validate a predictive model? [Електронний ресурс]. – Режим доступу: https://www.aspexit.com/en/how-to-validate-a-predictive-model/ – Дата доступу: листопад 2020.
27. Richards M. Software architecture patterns. – 1005 Gravenstein Highway North, Sebastopol, CA 95472 : O'Reilly Media, Incorporated, 2015. – Т. 4.
28. Dey T. A Comparative Analysis on Modeling and Implementing with MVC Architecture //IJCA Proceedings on International Conference on Web Services Computing (ICWSC). – 2011. – Т. 1. – С. 44-49.
29. Гамма Э. и др. Приемы объектно-ориентированного проєктирования. – " Издательский дом"" Питер""", 2013.
30. Tsanas A., Xifara A. Accurate quantitative estimation of energy performance of residential buildings using statistical machine learning tools //Energy and Buildings. – 2012. – Т. 49. – С. 560-567.
31. Linear Models. [Електронний ресурс]. – Режим доступу: https://scikit-learn.org/stable/modules/linear_model.html. – Дата доступу: березень 2021.

ДОДАТКИ

Додаток 1

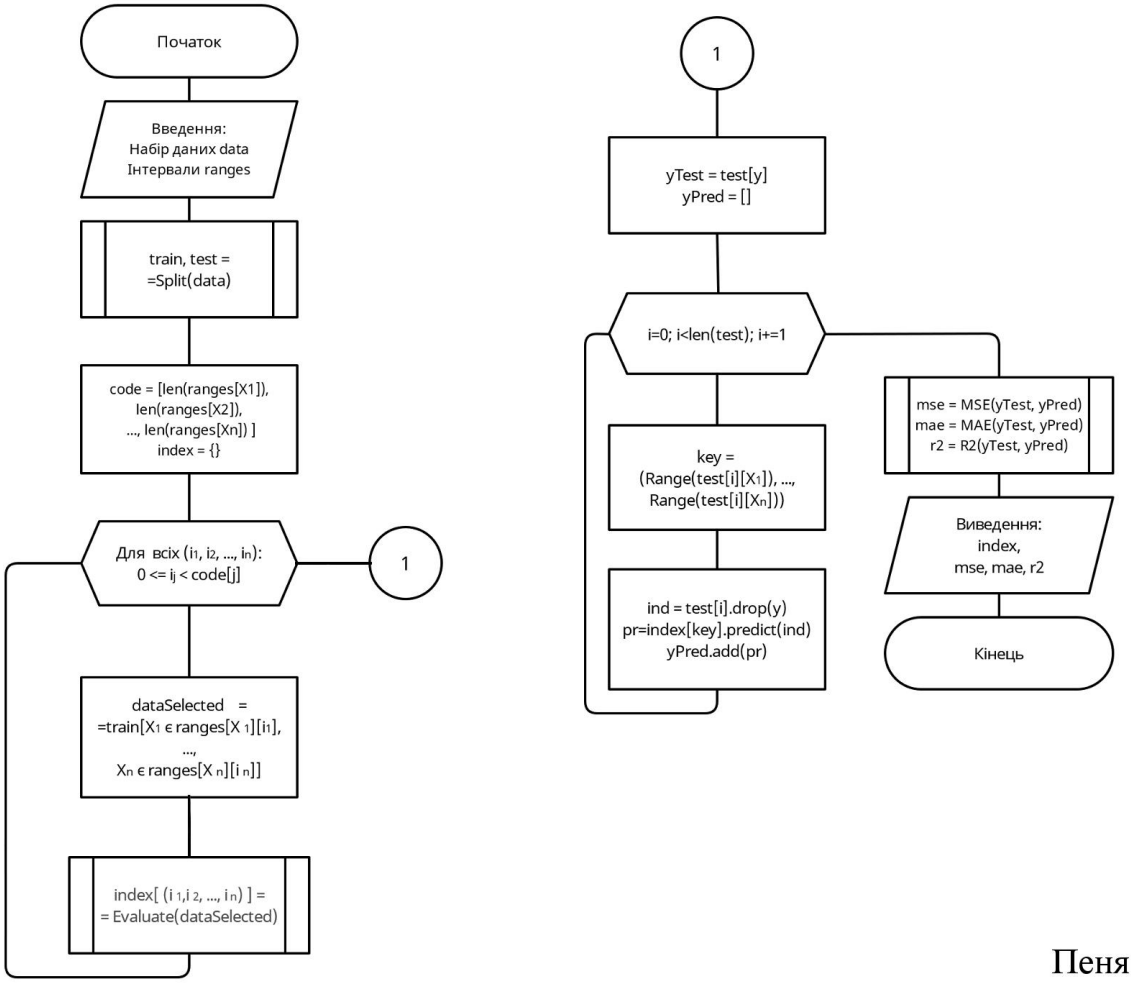
Копії графічних матеріалів

Алгоритм кластеризації, що враховує статистичні співвідношення



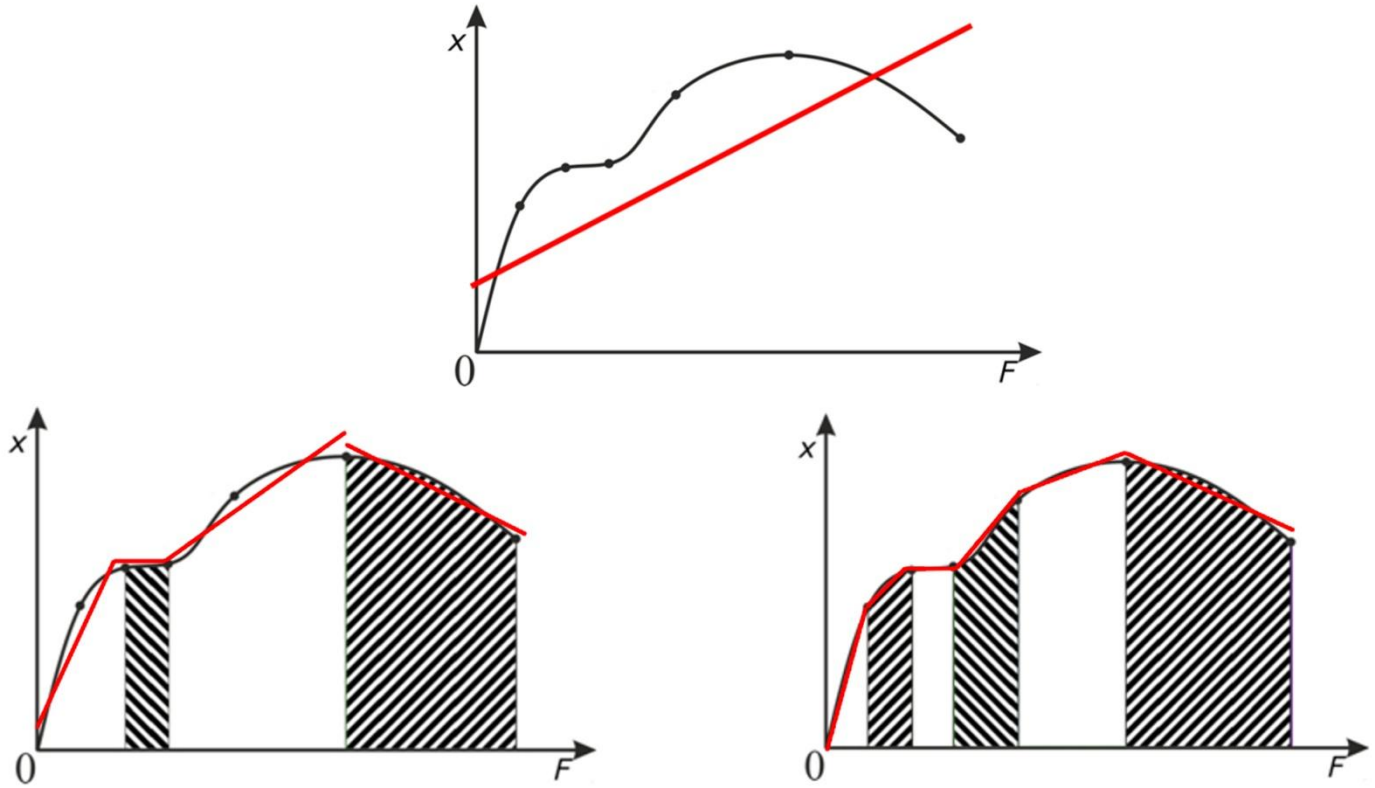
Пеня Олександр
КП-91мн

Алгоритм побудови та оцінки моделі



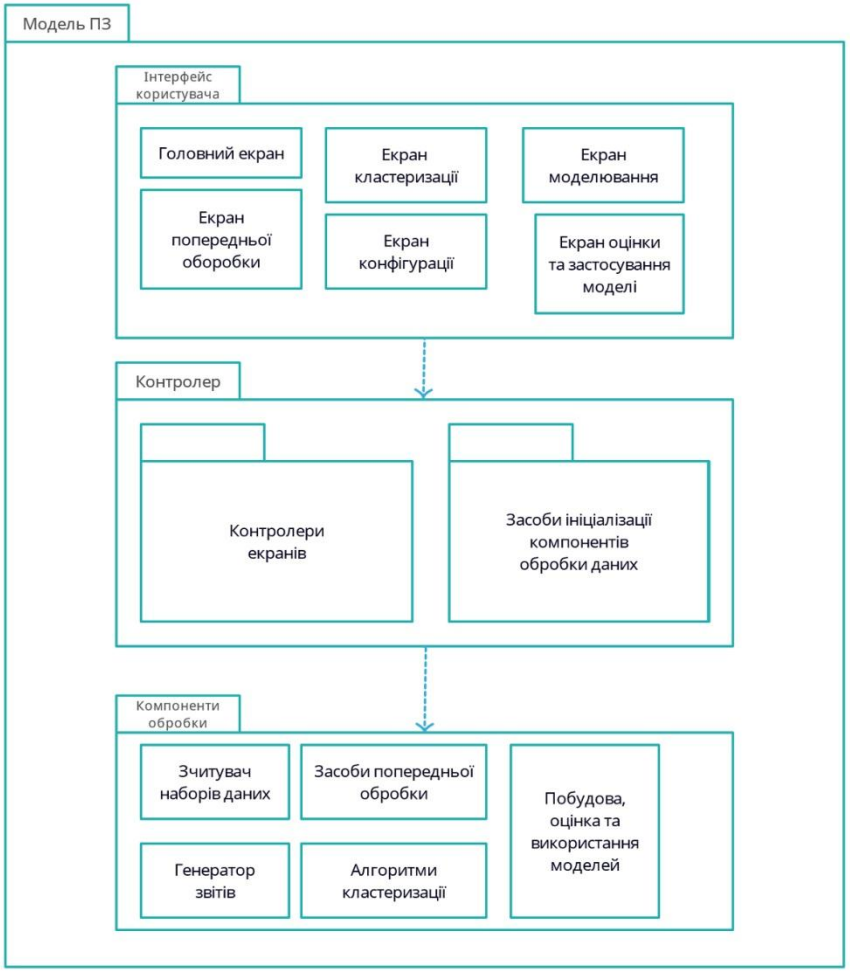
Пеня Олександр
КП-91мн

Використання одного і декількох наближень в різних станах



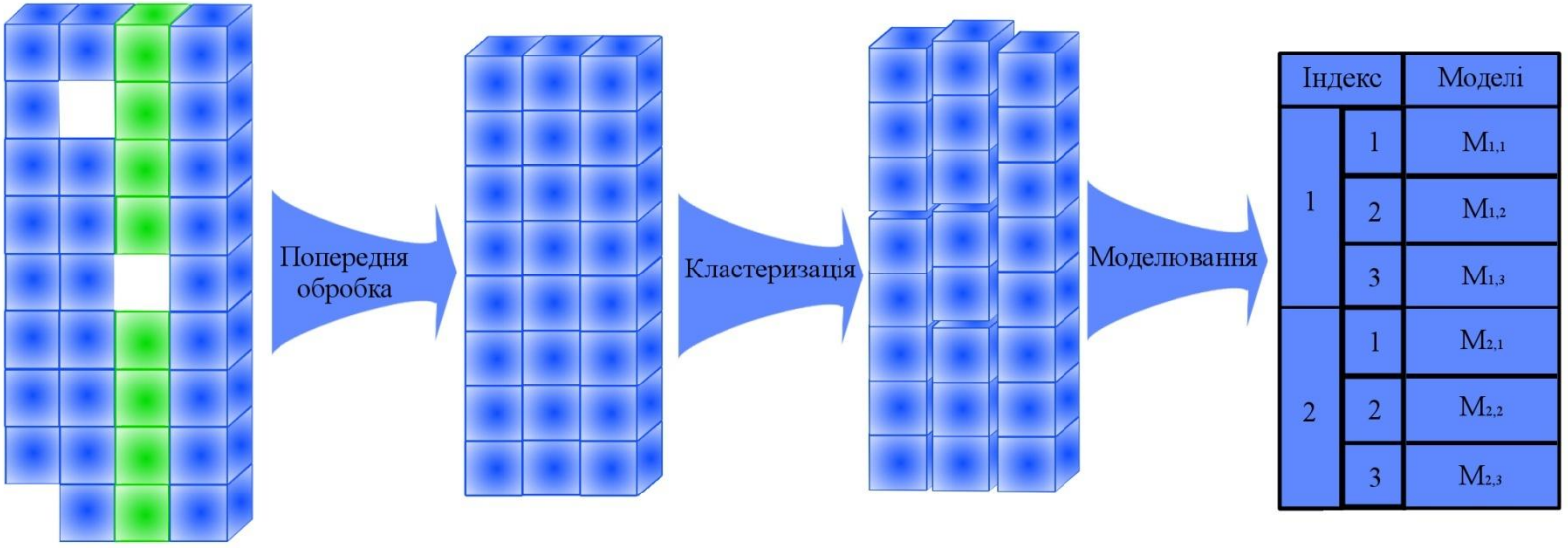
Пеня Олександр
КП-91мн

Діаграма моделі програмного забезпечення



Пеня Олександр
КП-91мн

Схема етапів методу



"Сирі" дані

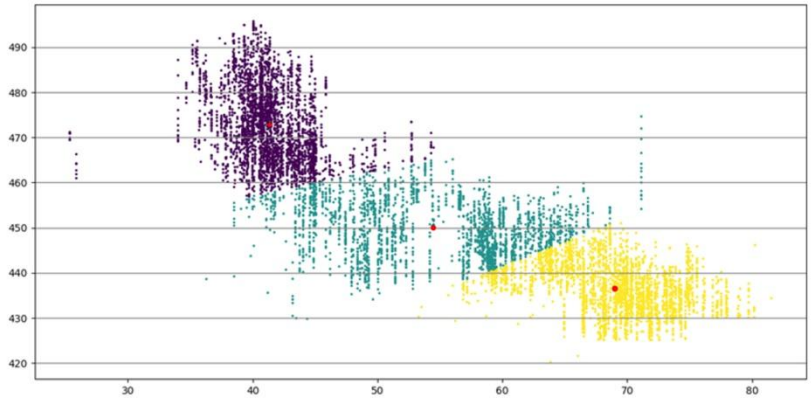
Структуровані дані

Інтервали простору

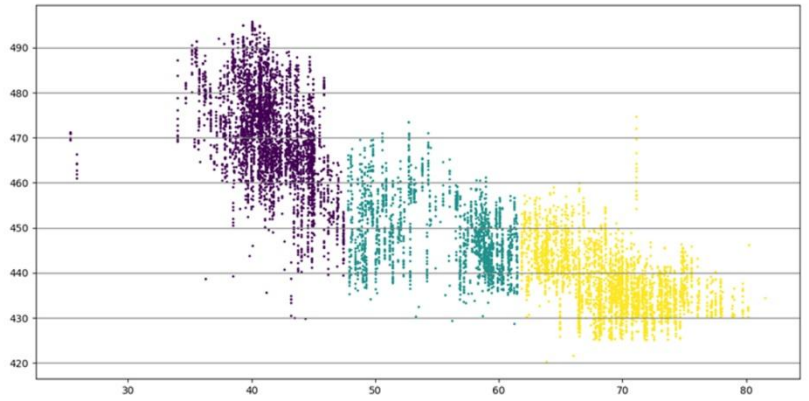
Моделі на інтервалах

Пеня Олександр
КП-91мн

Однозначний поділ простору



Інтервал визначається як залежною,
так і незалежною величиною



Інтервал визначається тільки
незалежною величиною

Пеня Олександр
КП-91мн

Додаток 2

Копія презентації

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО”

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

КАФЕДРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ КОМП’ЮТЕРНИХ СИСТЕМ

Алгоритмічно-програмний метод агрегації даних цифрових двійників

Виконав: Пеня Олександр

Науковий керівник: к.т.н., доцент, Сулема Є. С.

Київ – 2021

ОСНОВНІ ПОНЯТТЯ

- Мультимодальні дані – це дані, які представляють інформацію про деяку сутність або явище з різних боків (модальності).
- Поєднання даних (data fusion) – аналіз декількох наборів даних, таким чином, щоб вони могли взаємодіяти, інформувати та впливати один на одного для отримання більш точної, узгодженої або корисної інформації.
- Цифровий двійник – цифрове представлення фізичного об'єкта або процесу.

АКТУАЛЬНІСТЬ ДОСЛІДЖЕННЯ

- Оптимізація процесів
- Прийняття рішень
- Дослідження складних систем
- Підвищення безпеки

ПОСТАНОВКА ЗАДАЧІ

Задано повний узгоджений набір мультимодальних даних про деякий об’єкт, процес або систему.

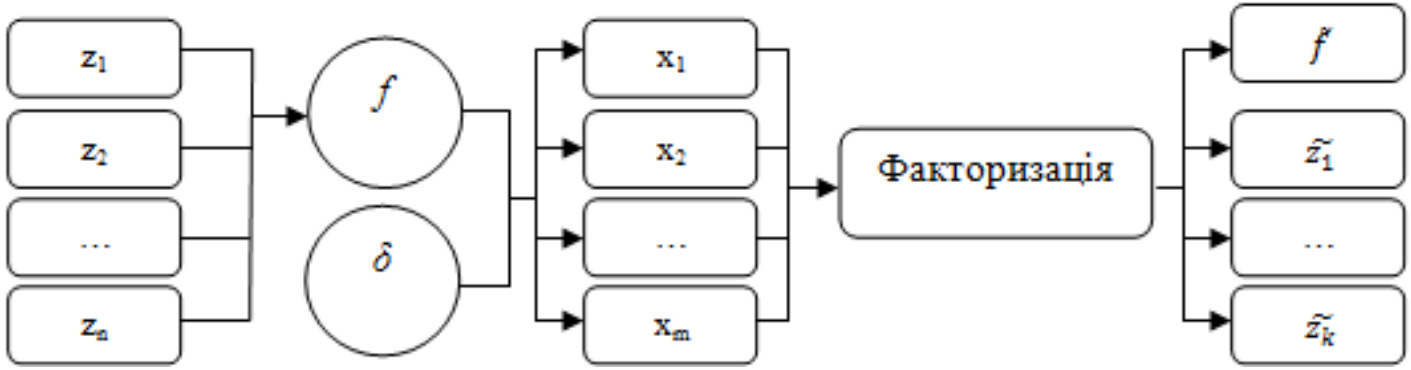
Необхідно побудувати цифрову модель системи за цим набором.

ІСНУЮЧІ МЕТОДИ

- Методи факторизації
- Імовірнісні підходи (Баєсівська модель)
- Методи машинного навчання
- Методи, що базуються на моделях

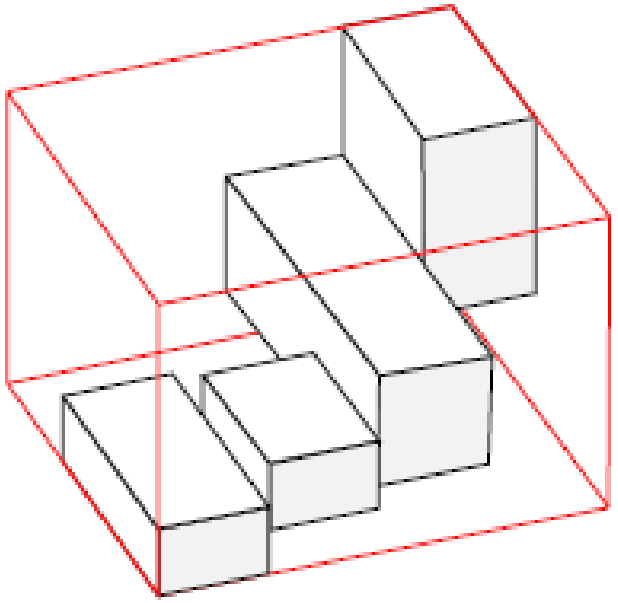
МЕТОДИ ФАКТОРИЗАЦІЇ

$$x_{ij} = \sum_{k=1}^{n} a_{ik} s_{jk}, \Leftrightarrow X = \sum_{k=1}^{n} \bar{a}_k \bar{s}_k^T = AS^T.$$



МЕТОДИ ПОЄДНАНОЇ ФАКТОРИЗАЦІЇ

- Аналіз незалежних векторів
- Паралельний факторний аналіз
- Поєднаний аналіз головних компонент



МЕТОДИ МАШИННОГО НАВЧАННЯ

- Штучні нейронні мережі
- Генетичні алгоритми
- Комбінований метод

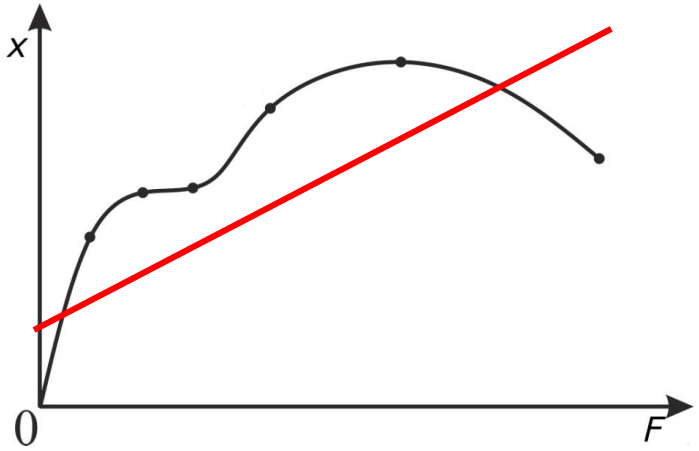
МЕТОДИ, ЩО БАЗУЮТЬСЯ НА МОДЕЛЯХ

Як приклад, багатовимірна лінійна регресія

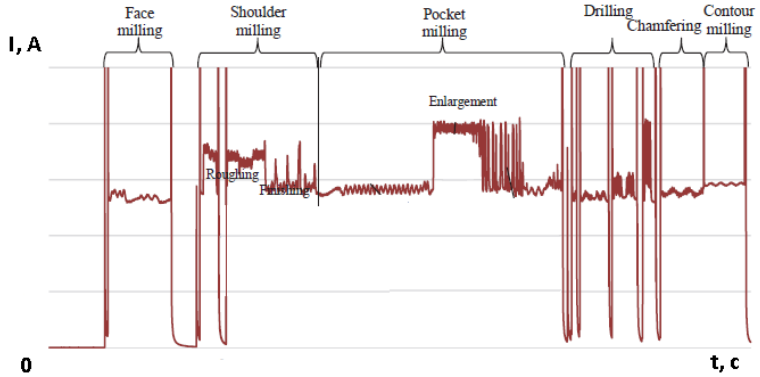
$$y = \sum_{i=1}^{n} a_i x_i + a_0$$

СКЛАДНІСТЬ І СТАНИ СИСТЕМ

Розглянуті методи побудови цифрових двійників мають значний недолік - використання однієї лінійної моделі для всього набору даних.



Деформаційна характеристика

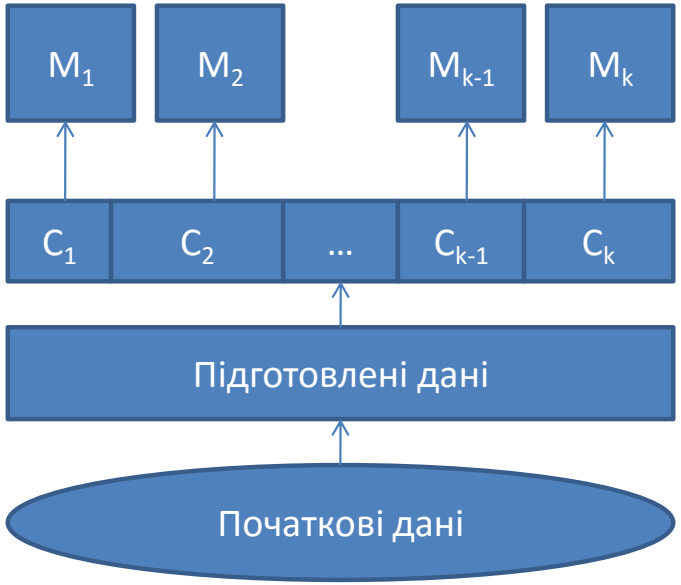


Робота станка з ЧПУ

НОВИЙ МЕТОД ПОБУДОВИ ЦИФРОВИХ ДВІЙНИКІВ

Для вирішення проблем розглянутих підходів пропонується метод, який складається з абстрактних етапів:

1. Підготовка.
2. Кластеризація.
3. Апроксимація на кластерах.



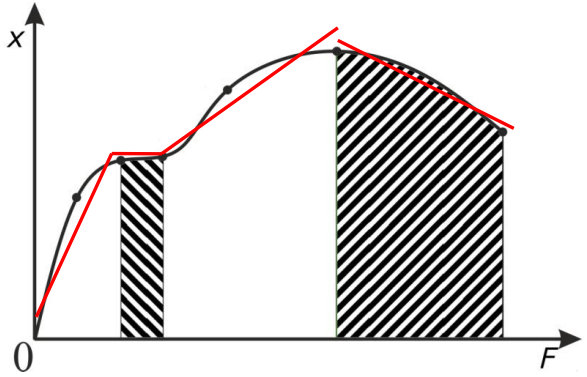
ПІДГОТОВКА ДАНИХ

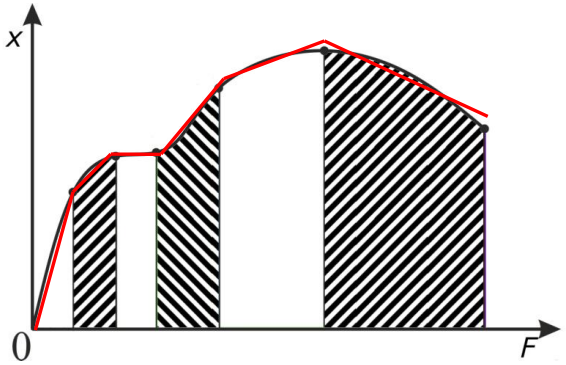
Обробка вихідного набору даних може включати:

- Відновлення відсутніх значень або вилучення неповних кортежів.
- Групування та видалення тотожних кортежів.
- Фільтрація.
- Доповнення даних.
- Скорочення даних.

КЛАСТЕРИЗАЦІЯ

Кластеризація дозволяє виділити окремі стани системи та розглядати їх незалежно щоб знайти більш точні моделі у кожному випадку.





АПРОКСИМАЦІЯ НА КЛАСТЕРАХ

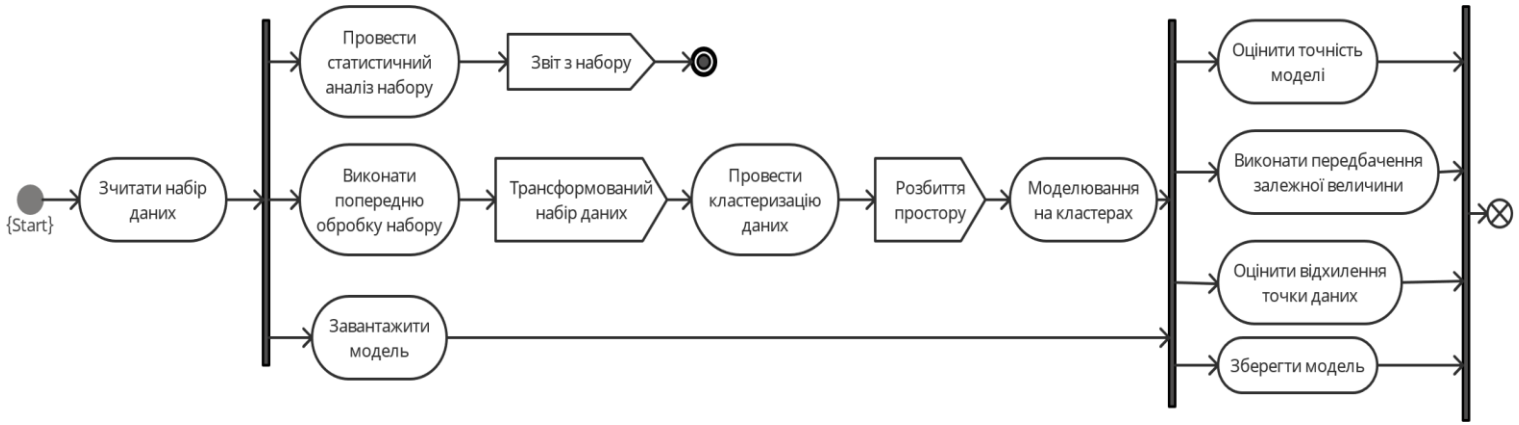
Для кожного кластеру C_i обирається своя модель представлення. Після цього виконується обчислення параметрів моделі на даних, що входять в даний кластер.

АНАЛІЗ ОТРИМАНОГО ЦИФРОВОГО ДВІЙНИКА

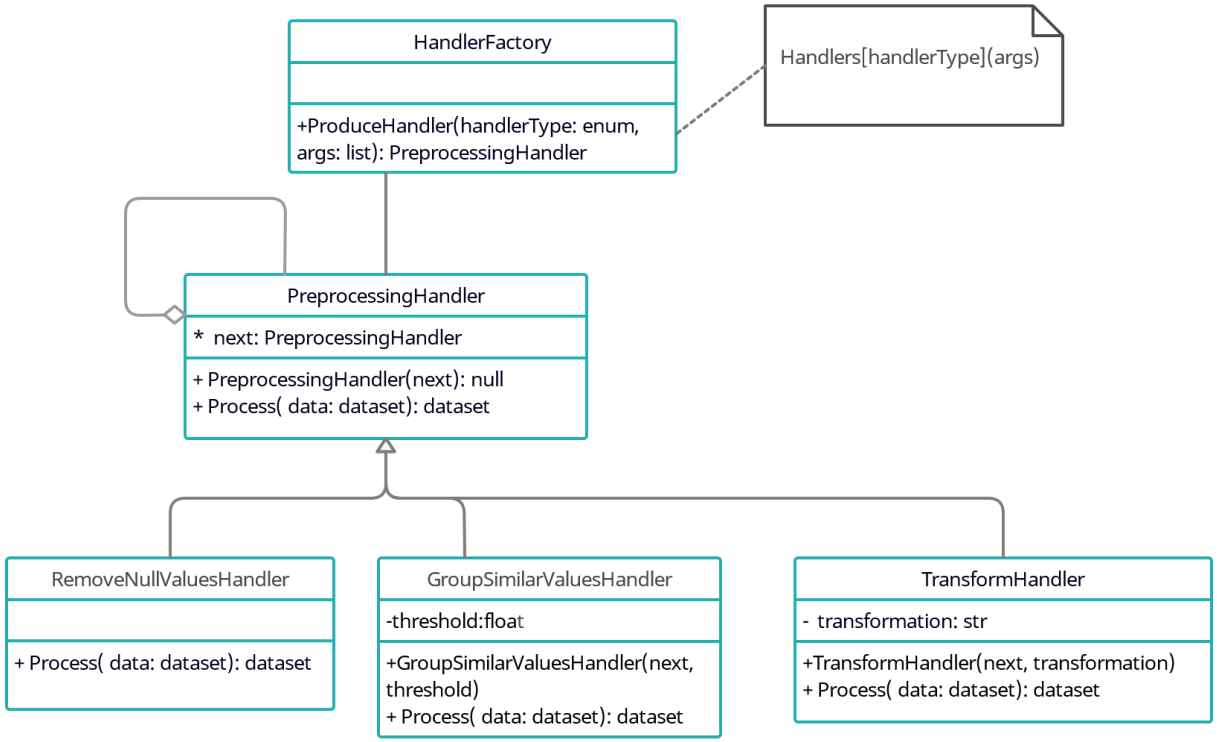
Після завершення обчислень потрібно виконати аналіз отриманої моделі.

Окрім верифікації на тестових даних необхідно оцінити кластери та моделі: з центрів та околів кластерів виявити відповідність станам системи та проаналізувати параметри отриманих моделей.

СЦЕНАРІЇ РОБОТИ З ПЗ



ПОПЕРЕДНЯ ОБРОБКА

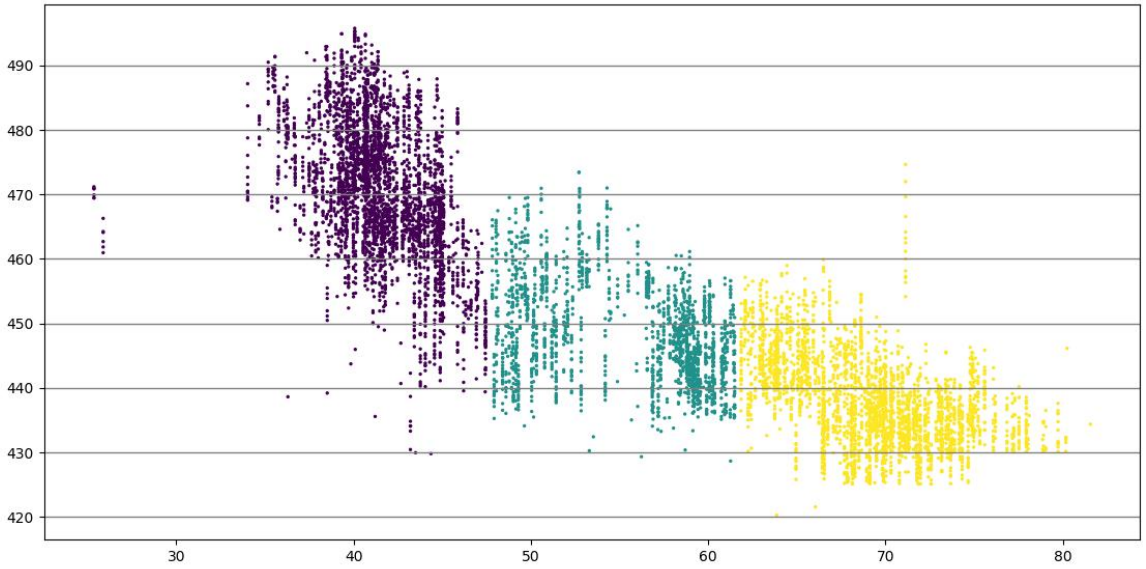


ПОБУДОВА МОДЕЛІ


```
class Model(object):
    def fit(self, dataset: pd.DataFrame, ranges: dict, independent: list, dependent: str):
        self._ranges = ranges
        self._index.clear()
        divisions = indexes = [list(range(len(l))) for l in ranges.values()]
        for combination in itertools.product(*divisions):
            self._index[combination] = self._fit_model(dataset, combination, independent,
    dependent)


    def predict(self, points: pd.DataFrame):
        points_copy = points.copy()
        points_copy["range_comb"] = points_copy.apply(lambda row: self._find_range(row),
    axis=1)
        points_copy["pred"] = points_copy.apply(lambda row: self._predict_row(row), axis=1)
        return points_copy["pred"].to_numpy(dtype=np.float32)
```

КЛАСТЕРИЗАЦІЯ

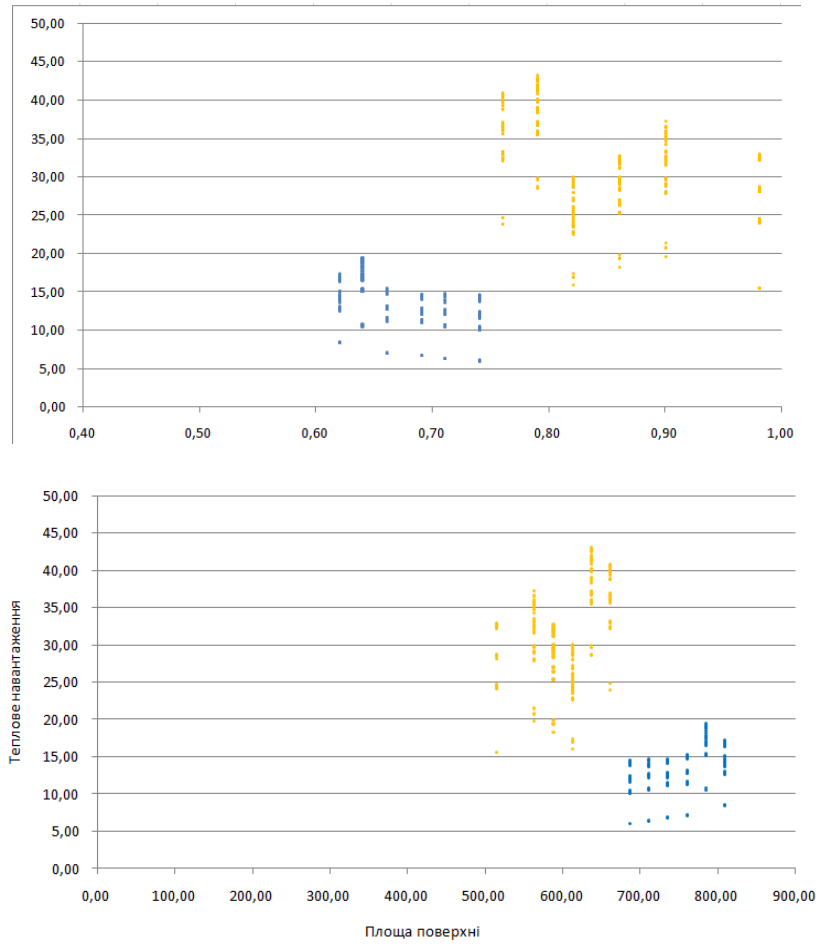


РЕЗУЛЬТАТИ ЗАСТОСУВАННЯ (1)

Незалежні величини	MSE	
	Множинна регресія	Застосування методу
V	72,20	71,01
AT	29,28	29,28
AT, V	24,13	22,59
AT, V, RH, AP	33,72	30,52

Незалежні величини	R^2	
	Множинна регресія	Застосування методу
V	0,75	0,79
AT	0,93	0,93
AT, V	0,95	0,91
AT, V, RH, AP	0,91	0,83

КЛАСТЕРИЗАЦІЯ



РЕЗУЛЬТАТИ ЗАСТОСУВАННЯ (2)

Незалежні величини	MSE		
	Множинна регресія	Кластеризація X1, X2	Кластеризація X1, X2, X3
Всі модальності	9.89	4.78	5.35
Неперервні модальності	9.78	4.77	

Незалежні величини	R^2		
	Множинна регресія	Кластеризація X1, X2	Кластеризація X1, X2, X3
Всі модальності	0.89	0.95	0.95
Неперервні модальності	0.89	0.95	

ПЕРЕВАГИ ТА НЕДОЛІКИ МЕТОДУ

Переваги:	Недоліки:
+ більша точність; + гнучкість; + можливість розподіленого виконання.	- складність: обчислювальна; інтерпретація; - статичність.

НАУКОВА НОВИЗНА

Наукова новизна полягає в тому, що було запропоновано метод поєднання мультимодальних даних, який дозволяє отримати більшу точність моделювання за рахунок використання декількох моделей в різних станах системи і можливості адаптувати реалізації його етапів до задачі.

ВИСНОВКИ І ПОДАЛЬША РОБОТА

Запропоновано метод побудови цифрових двійників за набором мультимодальних даних.

Подальші дослідження направлені на:

- формулювання способів визначення ефективних реалізацій етапів та їх параметрів;
- виправлення недоліків методу;
- покращення ПЗ.

Дякую за увагу

Додаток З

Лістинг програми

Лістинг 1. Зчитування набору даних (Reader.py)

```
import pandas as pd
from lxml import objectify
import os


class DataReader(object):
    def __init__(self):
        self.df = None

    def read(self, path: str, **kwargs):
        if os.path.isfile(path):
            if path.endswith(".csv"):
                self.df = pd.read_csv(path, **kwargs)
            elif path.endswith(".xml"):
                self.df = self._read_xml(path, **kwargs)
            elif path.endswith(".json"):
                self.df = pd.read_json(path, **kwargs)
            else:
                raise ValueError("Unknown file format")
            return self.df
        else:
            raise FileNotFoundError("No such file exist")

    def _read_xml(self, path: str, **kwargs):
        xml_data = objectify.parse(path)
        root = xml_data.getroot()

        data = []
        columns = []
        for i in range(len(root.getchildren())):
            child = root.getchildren()[i]
            data.append([subchild.text for subchild in
child.getchildren()])
            columns.append(child.tag)

        df = pd.DataFrame(data).T
        df.columns = columns
        return df

    def rename_columns(self, renames):
        self.df = self.df.rename(renames)
        return self.df

    def select_columns(self, columns):
        self.df = self.df[columns]
        return self.df
```

Лістинг 2. Попередня обробка (Prepocessing.py)

```
import pandas as pd
import numpy as np


class PreprocessingHandler(object):
    def __init__(self, next_handler=None):
        self.next = next_handler

    def process(self, data: pd.DataFrame):
        return data

    def _base_process(self, data: pd.DataFrame):
        if self.next is not None:
            data = self.next.process(data)
        else:
            data = data.reset_index()
        return data

    def get_parameters(self):
        return {}


class RemoveEmptyEntries(PreprocessingHandler):
    def __init__(self, next_handler=None):
        super().__init__(next_handler)

    def process(self, data: pd.DataFrame):
        data = data.dropna()
        return self._base_process(data)


class RemoveDuplicateEntries(PreprocessingHandler):
    def __init__(self, next_handler=None):
        super().__init__(next_handler)

    def process(self, data: pd.DataFrame):
        data = data.drop_duplicates()
        return self._base_process(data)


class Transform(PreprocessingHandler):
    def __init__(self, transform_expression: str, next_handler=None):
        super().__init__(next_handler)
        self.expression = transform_expression

    def process(self, data: pd.DataFrame):
        eq_split = self.expression.find("=")
        transformed_column = "'" + self.expression[:eq_split].strip()
+ "'"
        transform_formula = self.expression[eq_split+1:]
        for column in data.columns():
            transform_formula = transform_formula.replace(column,
"data['" + column + "']")
        data[transformed_column] = eval(transform_formula)

        return self._base_process(data)

    def get_parameters(self):
        return {"transform_expression": ("string", "Вираз
перетворення")}
```

Лістинг 3. Кластеризація (Clustering.py)

```
import pandas as pd
import numpy as np
from sklearn.cluster import KMeans


class Clustering(object):
    def __init__(self):
        pass

    def cluster(self, dataset: pd.DataFrame, independent: str,
prefered_modalities: list = None):
        pass


class CorrelationCoefficientClustering(Clustering):
    def __init__(self, step, threshold):
        super().__init__()
        self._step = step
        self.correlation_threshold = threshold

    def cluster(self, dataset: pd.DataFrame, independent: str,
prefered_modalities: list = None):
        if prefered_modalities is None:
            prefered_modalities = list(dataset.columns())
            prefered_modalities.remove(independent)

        ranges = {}
        for modality in prefered_modalities:
            ranges[modality] = self._cluster_modality(dataset,
independent, modality)

        return ranges

    def _cluster_modality(self, dataset: pd.DataFrame, independent:
str, modality: str):
        ranges = []
        partial = dataset[[independent,
modality]].sort_values(by=[modality], ascending=True)

        min_val = partial[modality].min()
        max_val = partial[modality].max()
        start = min_val
        final = max_val
        end = min_val + self._step * (max_val - min_val)

        while end < final:
            # estimate around a given range
            correlation = self._range_correlation(partial,
independent, modality, start, end)

            # estimate around larger ranges
            correlation1 = self._range_correlation(partial,
independent, modality, start, end + self._step * (max_val - min_val) / 2)
            correlation2 = self._range_correlation(partial,
independent, modality, start, end + self._step * (max_val - min_val))

            # choose appropriate range to examine

            if correlation1 >= correlation:
                if correlation2 >= correlation:
                    end = end + self._step * (max_val - min_val)
```

```
                    else:
                        end = end + self._step * (max_val - min_val) / 2
                else:
                    if correlation2 >= correlation:
                        #correlation3 = self._range_correlation(partial,
independent, modality, start, end + self._step * (max_val - min_val) * 2)
                        #if correlation3 >= correlation:
                            #end = end + self._step * (max_val - min_val)
                        #else:
                        if end - (max_val - min_val) < self._step *
(max_val - min_val):
                            end = max_val
                        ranges.append((start, end))
                        start = end
                        end = end + self._step * (max_val - min_val)
                    else:
                        if end - (max_val - min_val) < self._step *
(max_val - min_val):
                            end = (max_val - min_val)
                        ranges.append((start, end))
                        start = end
                        end = end + self._step * (max_val - min_val)

            return ranges

        def _range_correlation(self, dataset: pd.DataFrame, independent:
str, modality: str, start: float, end: float):
            partial = dataset.loc[(start <= dataset[modality]) &
(dataset[modality] <= end)]
            correlation = partial[modality].corr(partial[independent])
            if np.isnan(correlation):
                correlation = 0
            return abs(correlation)


    class KMeansCorrelationClustering(Clustering):
        def __init__(self, n_clusters, corr_range=0.1, overlap=0.125):
            super().__init__()
            self._n = n_clusters
            self._range = corr_range
            self._overlap = overlap

        def cluster(self, dataset: pd.DataFrame, independent: str,
prefered_modalities: list = None):
            if prefered_modalities is None:
                prefered_modalities = list(dataset.columns())
                prefered_modalities.remove(independent)

            ranges = {}
            for modality in prefered_modalities:
                ranges[modality] = self._cluster_modality(dataset,
independent, modality)

            return ranges

        def _cluster_modality(self, dataset: pd.DataFrame, dependent:
str, modality: str):
            partial = dataset[[dependent,
modality]].sort_values(by=[modality], ascending=True)
            partial["corr"] = 0.0
            partial = self._append_range_correraltion(partial, dependent,
modality)
            kmeans = KMeans(init="random", n_clusters=self._n, n_init=10,
max_iter=300)
```

```
            kmeans.fit(partial.to_numpy(dtype=np.float32))
            ranges = self._unambiguous_split(kmeans.cluster_centers_,
dataset[modality].min(), dataset[modality].max())
            return ranges

        def _append_range_correraltion(self, dataset: pd.DataFrame,
dependent, modality):
            min_value = dataset[modality].min()
            max_value = dataset[modality].max()
            neighbourhood_range = self._range * (max_value - min_value)
            start = min_value
            end = min_value + neighbourhood_range
            while end < max_value:
                subset = dataset[modality >= start - neighbourhood_range
* self._overlap &
                                 modality <= end + neighbourhood_range *
self._overlap]
                subset["corr"] = subset[modality].corr(subset[dependent],
method="pearson")

                dataset = dataset.join(subset, on="index", how="left")
                start = end
                end += neighbourhood_range

            return dataset

        def _unambiguous_split(self, centers, min, max):
            ranges = []
            start = min
            end = 0
            for i in range(len(centers) - 1):
                end = (centers[i+1][0] - centers[i][0]) / 2.0
                ranges.append((start, end))
                start = end
            ranges.append((start, max))
            return ranges
```

Лістинг 4. Моделювання (Modeling.py)

```
import pandas as pd
import itertools
from sklearn import linear_model
import numpy as np


class Model(object):
    def __init__(self):
        self._ranges = {}
        self._index = {}

    def fit(self, dataset: pd.DataFrame, ranges: dict, independent:
list, dependent: str):
        self._ranges = ranges
        self._index.clear()
        divisions = [list(range(len(l))) for l in ranges.values()]
        for combination in itertools.product(*divisions):
            self._index[combination] = self._fit_model(dataset,
combination, independent, dependent)

    def _fit_model(self, dataset: pd.DataFrame, combination: tuple,
independent: list, dependent: str):
        partial_data = dataset
        i = 0
        for divided_modality in self._ranges.keys():
            partial_data =
partial_data.loc[(partial_data[divided_modality] >=
self._ranges[divided_modality][combination[i]][0]) &

(partial_data[divided_modality] <=
self._ranges[divided_modality][combination[i]][1])]
            i += 1

        train_x = partial_data[independent]
        train_y = partial_data[dependent]

        if len(train_x.index) > 0:
            regression = linear_model.LinearRegression()
            regression.fit(train_x, train_y)
            return regression
        else:
            return None

    def predict(self, points: pd.DataFrame):
        points_copy = points.copy()
        points_copy["range_comb"] = points_copy.apply(lambda row:
self._find_range(row), axis=1)
        points_copy["pred"] = points_copy.apply(lambda row:
self._predict_row(row), axis=1)
        return points_copy["pred"].to_numpy(dtype=np.float32)

    def _find_range(self, row):
        position = []
        for divided_modality in self._ranges.keys():

position.append(self._locate(self._ranges[divided_modality],
row[divided_modality]))
        return tuple(position)

    def _locate(self, ranges, value):
        i = 0
```

```
                for item in ranges:
                    if value <= item[1]:
                        break
                    i += 1
                return i

            def _predict_row(self, row):
                model = self._index[row["range_comb"]]
                values = row.drop(["range_comb",
"index"]).to_numpy(dtype=np.float32)
                return model.predict(values.reshape(1, -1))
```